L-3: SSH, Autolab, Variables, Types, scanf

Sandeep Nagar

Part-1

SSH, Autolab(pingala), pingala shell, Autograder

Part-2

Comments, Identifiers, Variables, Types, Constants, scanf, Controle Flow

L-3 Slides: https://cpro-iiit.github.io/docs/course_material/lectures/3/lec_3.pdf

Reference code: https://tinyurl.com/yuhchnuj

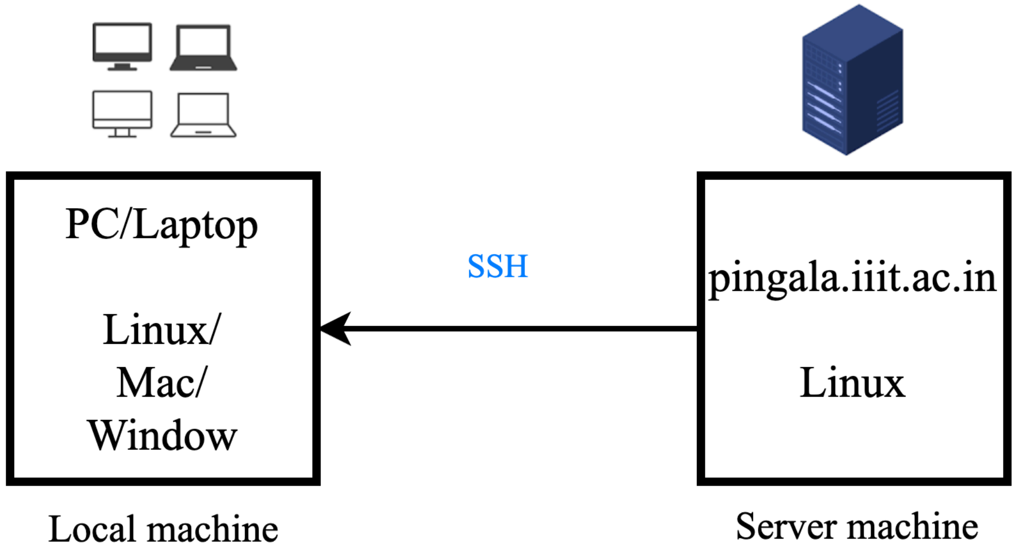
Programiz, web editor: https://tinyurl.com/bdd55vwn

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

What is SSH, and how do I use it?

```
ssh sandeep.nagar@pingala.iiit.ac.in
```

- Connects to pingala server (at IIIT with Linux OS with all programs required for the course installed).
- Why?: All students will work in the same environment (os same, programs same, etc.)



Log in over SSH

```
ssh user_name@pingala.iiit.ac.in
Enter your CAS password
```

- You can work on the remote machine using your local computer.
- You can edit, create, and copy files on the server.
- Submit assessments using your local machine to Autolab.

Autolab:

For automatic evaluation and grading of programs.

Two ways to submit for auto-grading:

- pingala shell: using ssh shell (prefered)
- GUI: user interface, using pingala.iiit.ac.in website

Questions about Autolab/ssh/pingala?

Running the Program on shell

1. Run gcc compiler to get the executable file `main`
   ```
   gcc main.c -o main
   ```
2. Run the executable `main`
   ```
   ./main
   ```

Comments for C:

- Whole-line comment
- Partial line comment
- Multiple line comment

```
// This is a whole-line comment
variable = 5; // this is partial line comment
/* and
comment
comment
..
*/
```

- Programiz, web editor: https://tinyurl.com/bdd55vwn

Identifiers:

- Unique names that are assigned to variables, structs, functions, and other entities.
- Allow us to name data and other objects in the program.
- Each identifier object in the computer is stored at a unique address.

Rules to create identifiers:

- First character must be alphabetical or underscore '_'
- Must contain only alphabetical characters, digits, or underscore
- The first 63 characters of an identifier are sufficient
- Can not duplicate a keyword

E.g. for identifiers

```
a                  // valid
my_name            // valid
_your_name_        // valid
_Bool              // valid
_bool              // valid but not same as _Bool
Student Name       // invalid
int                // not valid, int is a keyword
char               // not valid, char is a keyword
2_name             // invalid, starting with digit
I_am-Yoda          // invalid, '-' not allowed
```

Constants:

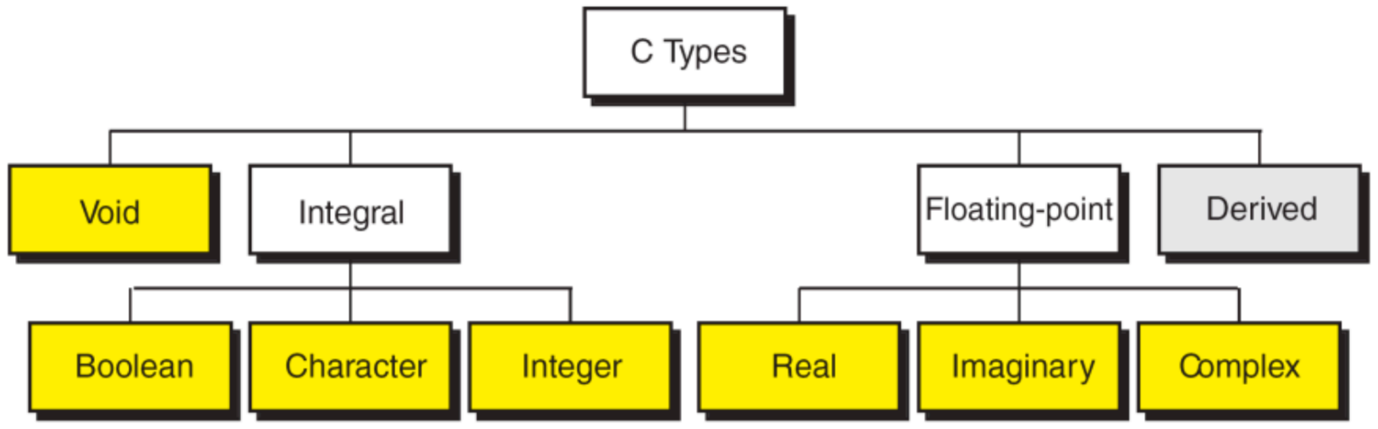
Constants are data values that can not be changed during the execution of a program. Like variables, constants have a type.

Constant types:

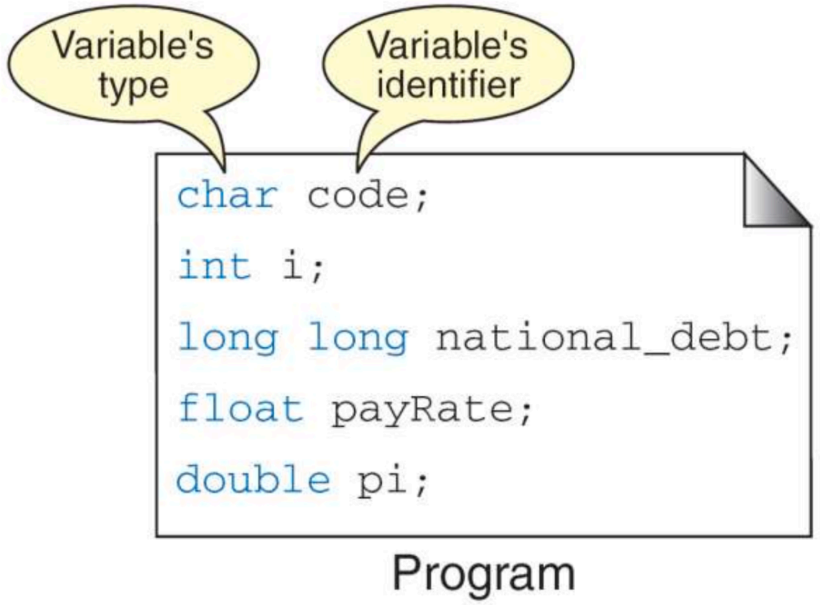
- **Boolean, character, integer, real, complex, and string constants.**

Variables:

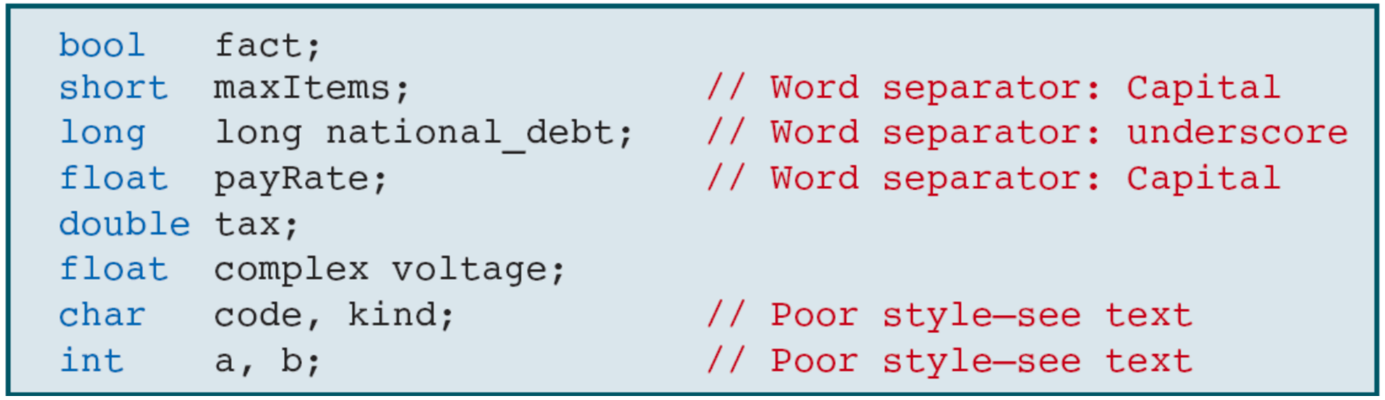
```
Void, Character, Integer
```



Variable Initialization:



```
char code;
int i;
long long national_debt;
float payRate;
double pi;
```

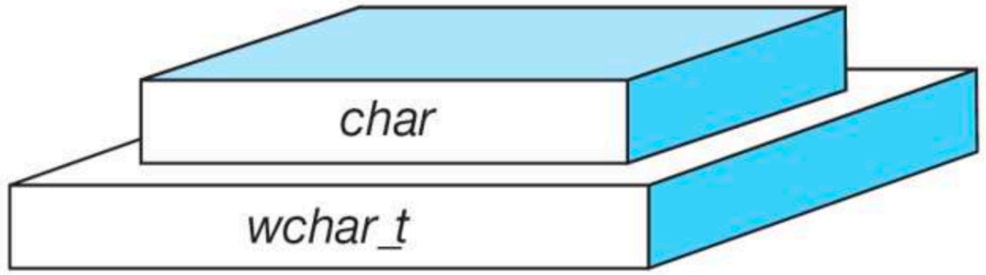


```
bool   fact;
short  maxItems;                  // Word separator: Capital
long   long national_debt;        // Word separator: underscore
float  payRate;                   // Word separator: Capital
double tax;
float  complex voltage;
char   code, kind;                // Poor style—see text
int    a, b;                      // Poor style—see text
```

Character Types:

```
// char, 1 byte (= 8 bit)

printf("%c", _char_)
```



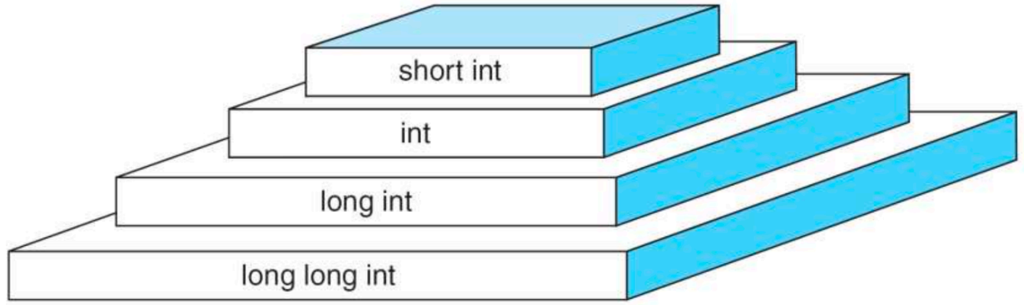
Integer Types:

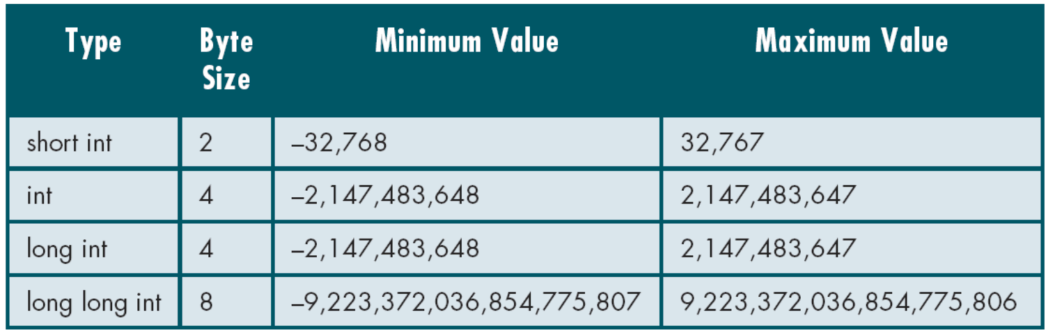
short, int, long, long long

- Size of integers

```
size of (short) ≤ size of (int) ≤ size of (long) ≤ size of (long long)

2 byte -> 4 byte = 4 byte -> 8 byte
```



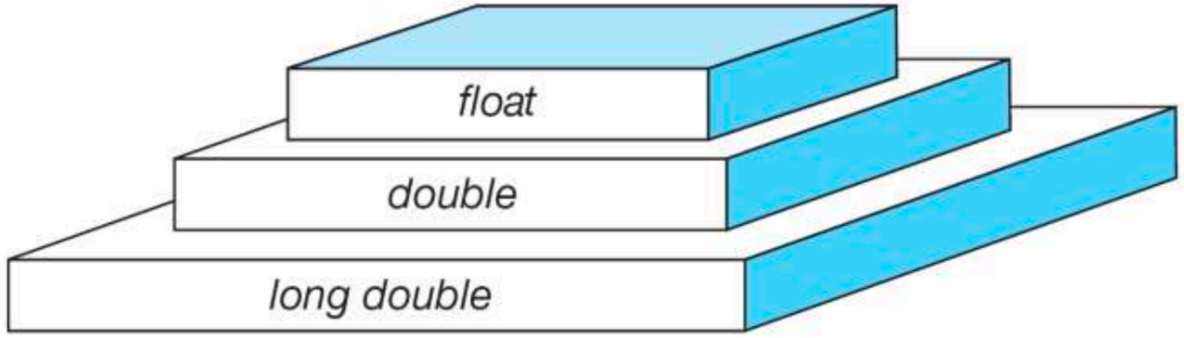


Type	Byte Size	Minimum Value	Maximum Value
short int	2	–32,768	32,767
int	4	–2,147,483,648	2,147,483,647
long int	4	–2,147,483,648	2,147,483,647
long long int	8	–9,223,372,036,854,775,807	9,223,372,036,854,775,806

Floating-point type:

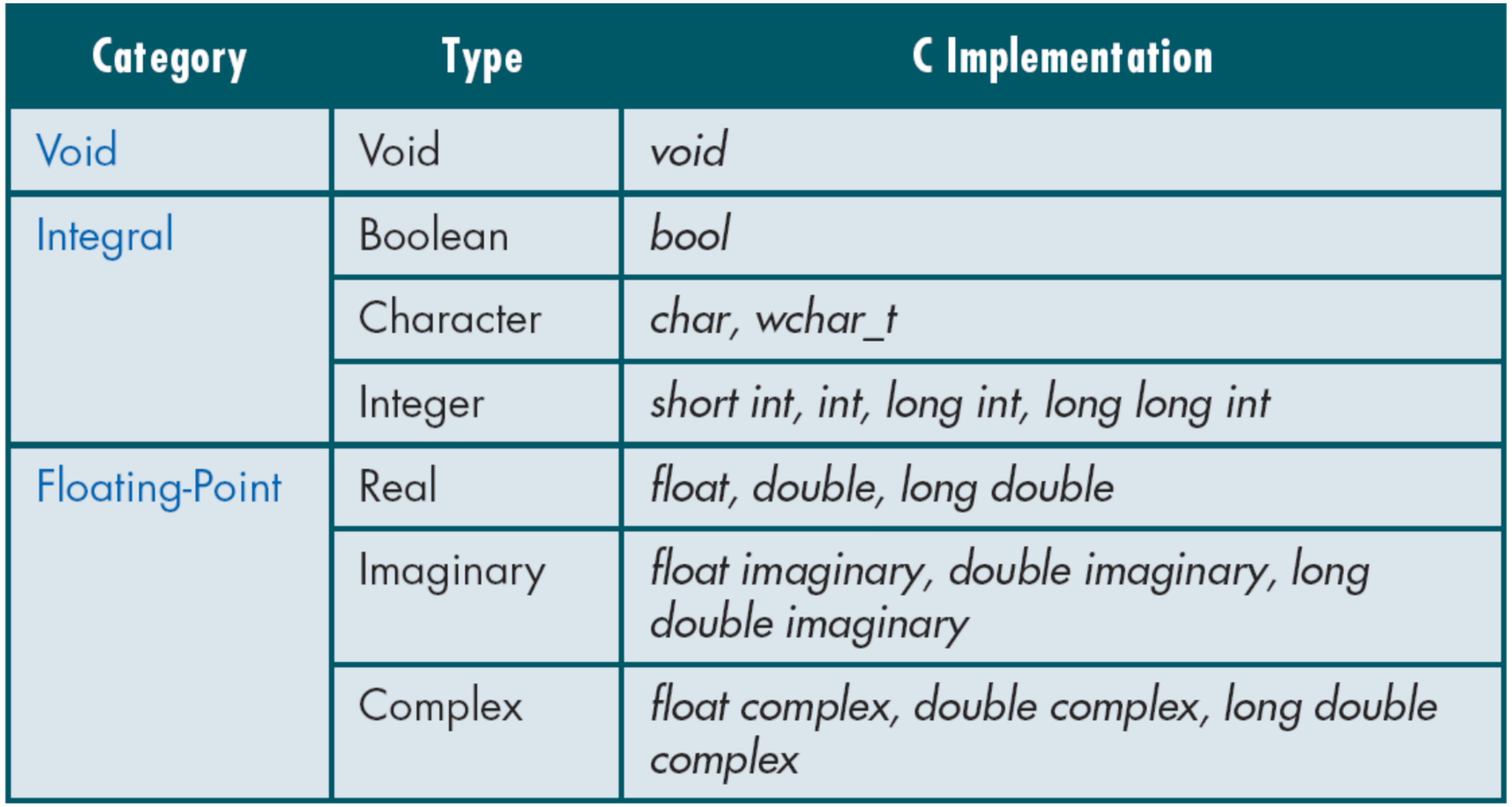
- **float, double, long double**

```
size of (float) ≤ size of (double) ≤ size of (long double)
4 byte -> 8 byte -> 16 byte
```



INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Type summary:



Category	Type	C Implementation
Void	Void	*void*
Integral	Boolean	*bool*
	Character	*char, wchar_t*
	Integer	*short int, int, long int, long long int*
Floating-Point	Real	*float, double, long double*
	Imaginary	*float imaginary, double imaginary, long double imaginary*
	Complex	*float complex, double complex, long double complex*

Type summary:

Conversion character	Discription	Example code

%d	For an integer in deciaml system	int m = 33;
		printf("%d", m);

%f	For a float type	float m_float = 33.33;
		printf("%f", m_float);

%c	For a character	char m_char = "C";
		printf("%c", m_char);

%s	For a string of characters	char m_string[4] = 'Cpro';
		printf("%s", m_string);

Symbolic names for control characters

- Some common control characters along with their symbolic names:

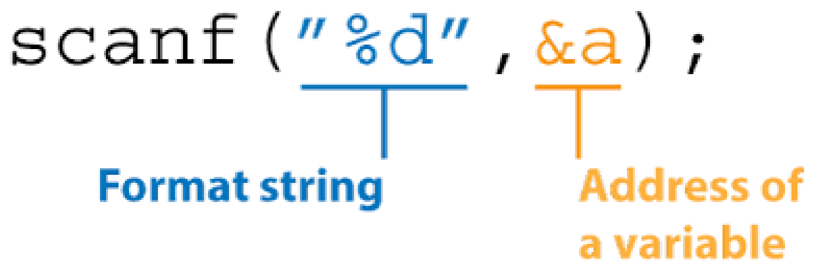
```
1.  Newline:           `\n`       printf("\n")
2.  Horizontal tab:    `\t`       printf("\t")
3.  Vertical tab:      `\v`       printf("\v")
4.  Backspace:         `\b`       printf("\b")
5.  Carriage return:   `\r`       printf("\r")
6.  Form feed:         `\f`       printf("\f")
7.  Alert (bell):      `\a`       printf("\a")
8.  Backslash:         `\\`       printf("\\")
9.  Single quote:      `\'`       printf("\'")
10. Double quote:      `\"`       printf("\"")
11. Question mark:     `\?`       printf("\?")
12. Null character:    `\0`       printf("\0")
```

scanf()

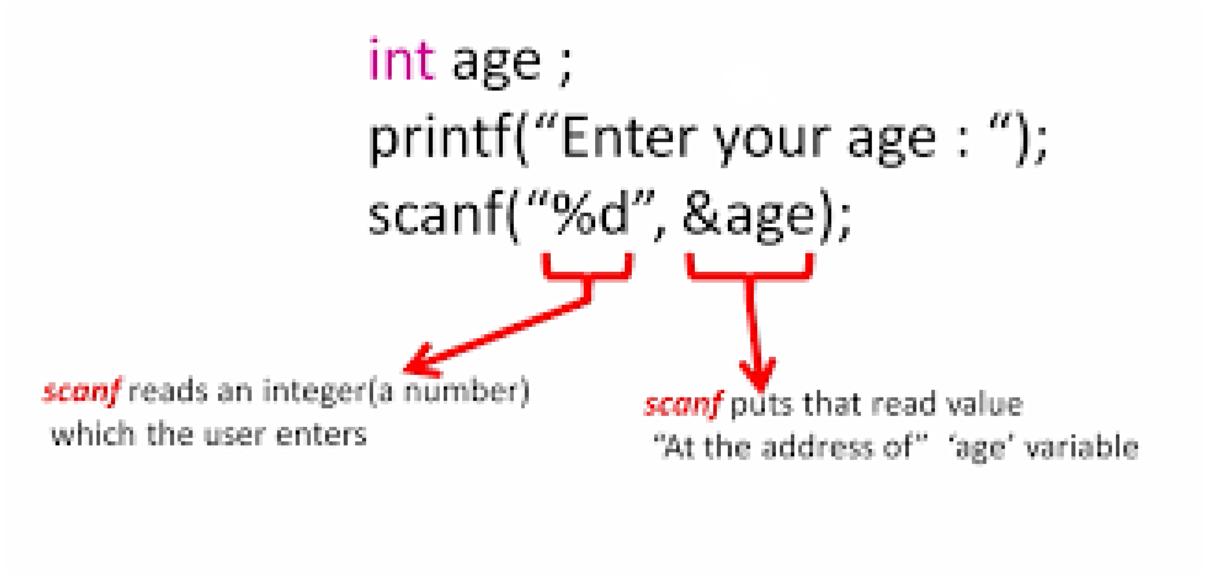
- Function reads data from the standard input stream stdin into the given locations.
- Reads `format-string` from left to right

int a = 5;

scanf("%d", &a);



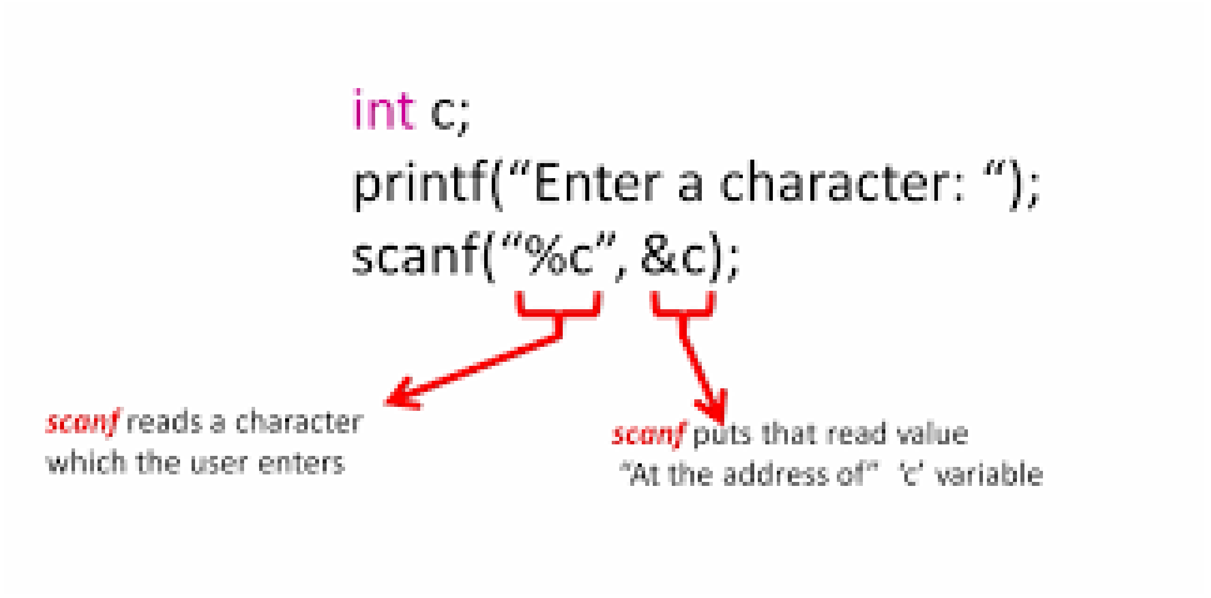
scanf()



```
int age ;
printf("Enter your age : ");
scanf("%d", &age);
```

scanf reads an integer(a number) which the user enters
scanf puts that read value "At the address of" 'age' variable

scanf()



```
int c;
printf("Enter a character: ");
scanf("%c", &c);
```

scanf reads a character which the user enters
scanf puts that read value "At the address of" 'c' variable

scanf()

Conversion character	Discription	Example code
%d	For an integer in deciaml system	scanf("%d", &a_int);
%f	For a float type	scanf("%f", &a_float);
%c	For a character	scanf("%c", &a_char);
%s	For a string of characters	scanf("%s", a_string);

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Control Flow

- Condition is an expression (or series of expressions)
  e.g. `n < 3 or x < y || z < y`
- Operators Precedence and Associativity: some operations are done before others when evaluating an expression.

```
Parentheses: ()                                         // first
Postfix operators: ++, --
Unary operators: +, -, !, ~, ++, --, (type)
Multiplicative operators: *, /, %
Additive operators: +, -
Relational operators: <, >, <=, >=
Equality operators: ==, !=
Logical AND operator: &&
Logical OR operator: ||
Assignment operators: =, +=, -= ... and so on   // last
```

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Associativity:

When expressions contain operators of the same precedence level, their evaluation order is determined.

- Left-Associative: operators are evaluated from left to right, `+`, `+`
  - e.g. `a + b - c` will first evaluate `a + b` and then subtract `c` from the result.
- Right-Associative: are evaluated from right to left, e.g. `=`
  - e.g. `a = b = c`, `c` is assigned to `b`, and then the resulting value of `b` is assigned to `a`.

Crucial for correctly interpreting and writing C programming expressions.

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Questions?

Reading

Next: Conditional Statements: if, else, while, switch, break, continue.

- Chapter 3: Computer Science: A Structured Programming Approach Using C Behrouz A. Forouzan, Richard F. Gilberg
- More about scanf : https://www.ibm.com/docs/en/i/7.4?topic=functions-scanf-read-data
- Programiz, web editor: https://tinyurl.com/bdd55vwn
- http://courses.washington.edu/mengr477/resources/Precedence.pdf