Google Developer Student Clubs

Embedded systems

Introduction to Embedded systems & C programming language

Youssef Abdelhakem
@YoussefAbdelhakem

Google Developer Student Clubs

Embedded systems

Outlines

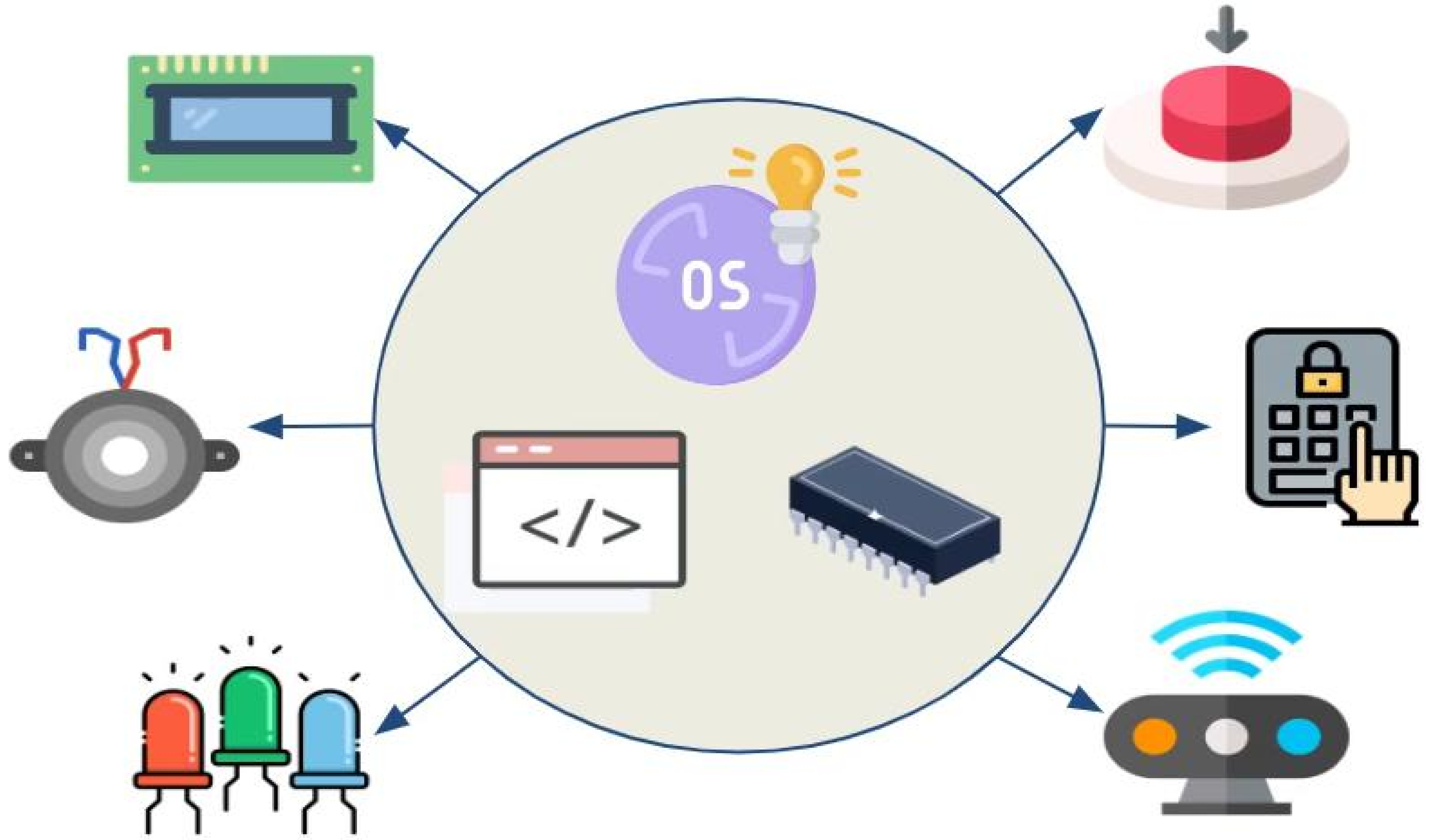
- What is embedded systems ?
- Embedded systems characteristics ?
- Embedded systems advantages and disadvantages ?
- Embedded systems applications ?

Embedded systems

What is embedded systems ?

- It's hardware controlled by software to preform specific and periodic functionality

- It may be real-time or not



Embedded systems

Embedded systems characteristics ?

- **Single-functioned**: repeated single functionality.
- **Tightly constrained**: small size, speed, low power consumption.
- **Reactive and Real time**: reacts to change in system environment.
- **Microprocessors based**: no embedded system without a microprocessors or a microcontroller.
- **Memory**: limited memory size.
- **Connected**: must be connected to input and output devices

Embedded systems

Embedded systems advantages and disadvantages ?

-Advantages

- Easily Customizable
- Low power consumption
- Low cost
- Enhanced performance

-Disadvantages

- High development effort
- Limited resources, memory, processing speed

Embedded systems

Embedded systems applications ?

- Automotive: Cruise control, light control, ABS, EBD,ESP,.. etc.
- Networking: Routers.
- Fintech: ATM, Point Of Sale, Vending machines,.. etc.
- Home appliances: Home automation, Air conditioners, microwave ovens, washing machines and dishwashers,... etc.
- Biomedical: Wearable devices, Teleradiology.
- Military: Missile targeting systems, command-and-control systems, electronic warfare.
- Consumer Electronics: MP3 players, television sets, mobile phones, video game Consoles , digital cameras, GPS receivers, printers,.. etc.

Google Developer Student Clubs

C programming language

Outlines

- What is C programming language.?
- Basic C program structure.
- Hello world in C.
- Variables in C.

C programming language

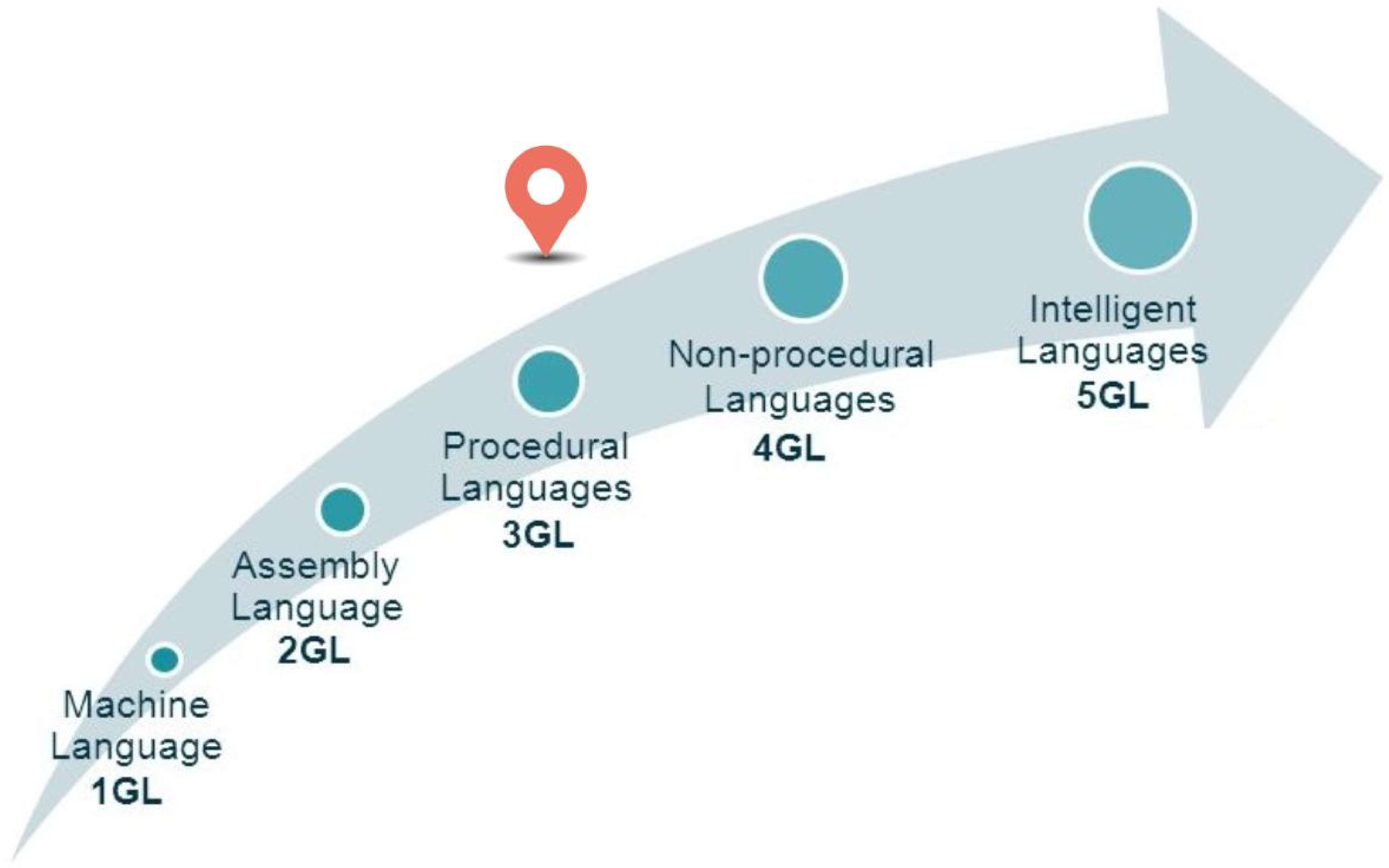
What is C programming language and why we learn it ?

- C is a general-purpose programming language created by Dennis Ritchie at the Bell Laboratories in 1972.

- C is strongly associated with UNIX, as it was developed to write the UNIX operating system.
- It is one of the most popular programming language in the world
- If you know C, you will have no problem learning other popular programming languages such as Java, Python, C++, C#, etc, as the syntax is similar
- C is very fast, compared to other programming languages, like Java and Python
- C is very versatile; it can be used in both applications and technologies

C programming language

Programming Generation Levels



Basic C program structure and Hello world.

```
#include <stdio.h>

int main() {

   /* my first program in C */

     // This C program to print Hello world

   printf("Hello, World! \n");

   return 0;

}
```

Google Developer Student Clubs

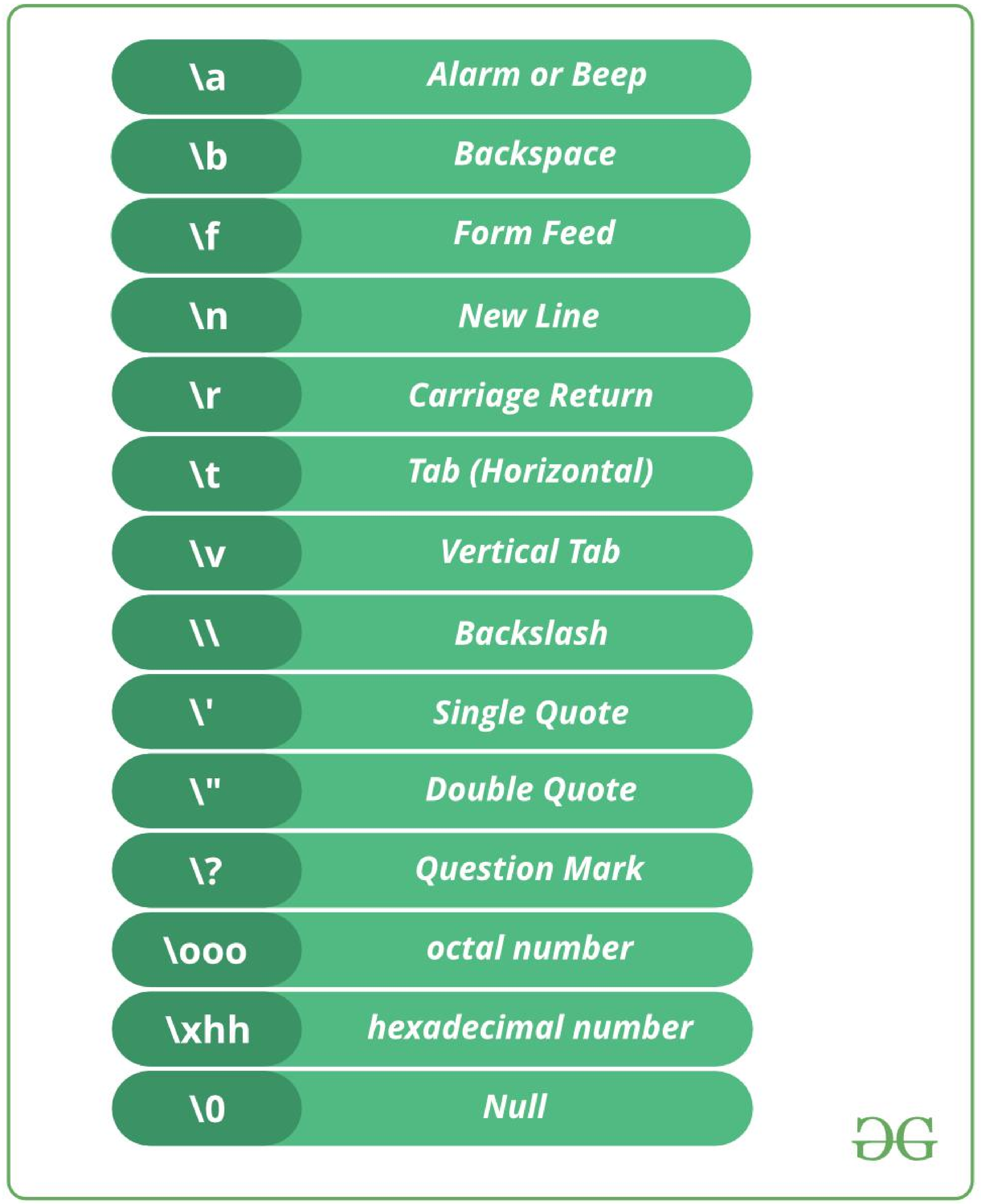
C programming language

Basic C program structure and Hello world.

1. # include <stdio.h> – This command is a preprocessor directive in C that includes all standard input-output functions
2. int main() – This is the line from where the execution of the program starts. The main() function starts the execution of any C program.
3. { (Opening bracket) – This indicates the beginning of any function in the program (Here it indicates the beginning of the main function).
4. /* some comments */ – Whatever is inside /*——-*/ are not compiled and executed; they are only written for user understanding or for making the program interactive by inserting a comment
5. printf("Hello World") –The printf() command is included in the C stdio.h library, which helps to display the message on the output screen.
6. return 0 –This command terminates the C program and returns a null value, that is, 0.
7. } (Closing brackets)- This indicates the end of the function. (Here it indicates the end of the main function)

C programming language

Escape Sequences in C



\a	*Alarm or Beep*
\b	*Backspace*
\f	*Form Feed*
\n	*New Line*
\r	*Carriage Return*
\t	*Tab (Horizontal)*
\v	*Vertical Tab*
\\	*Backslash*
\'	*Single Quote*
\"	*Double Quote*
\?	*Question Mark*
\ooo	*octal number*
\xhh	*hexadecimal number*
\0	*Null*

```
#include<stdio.h>
    int main()
{
        printf(“Youssef\n");
        printf(“Abdelhakem\t\n");
        printf(“Embedded systems\\");
    }
```

C programming language

Data types in C

- Primitive (Primary) Data Types : These data types store fundamental data used in the C programming.

1. **int** : Only integers, it`s with size : **4 Byte**
2. **long long** : Only integers, it`s with size : **8 Byte**
3. **float** : Decimals and integers, it`s with size : **4 Byte**
4. **double** : Decimals and integers, it`s with size : **8 Byte**
5. **char** : Symbols, it`s with size : **1 Byte**

- Derived and User Defined Data Types : These are made by collection or combination of primitive data types (Array ,Structure , Union , Enums)

C programming language

Signed Data types

Data type	Range	Format Specifier
1. **int : 4 Byte**	-2,147,483,648 to 2,147,483,647	%d-%i
2. **long : 8 Byte**	-(2^63) to (2^63)-1	%lld
3. **float : 4 Byte**	1.2E-38 to 3.4E+38	%f
4. **double : 8 Byte**	1.7E-308 to 1.7E+308	%lf
5. **char : 1 Byte**	-128 to 127	%c

C programming language

Unsigned Data types

Data type	Range	Format Specifier
1. **Unsigned int : 4 Byte**	-0 to 4,294,967,295	%u
2. **Unsigned long : 8 Byte**	0 to 18,446,744,073,709,551,615	%llu
3. **float : 4 Byte**	1.2E-38 to 3.4E+38	%f
4. **double : 8 Byte**	1.7E-308 to 1.7E+308	%lf
5. **Unsigned char : 1 Byte**	0 to 255	%c

C programming language

Variables in C

- Variable: placeholder helps you access data stored in memory

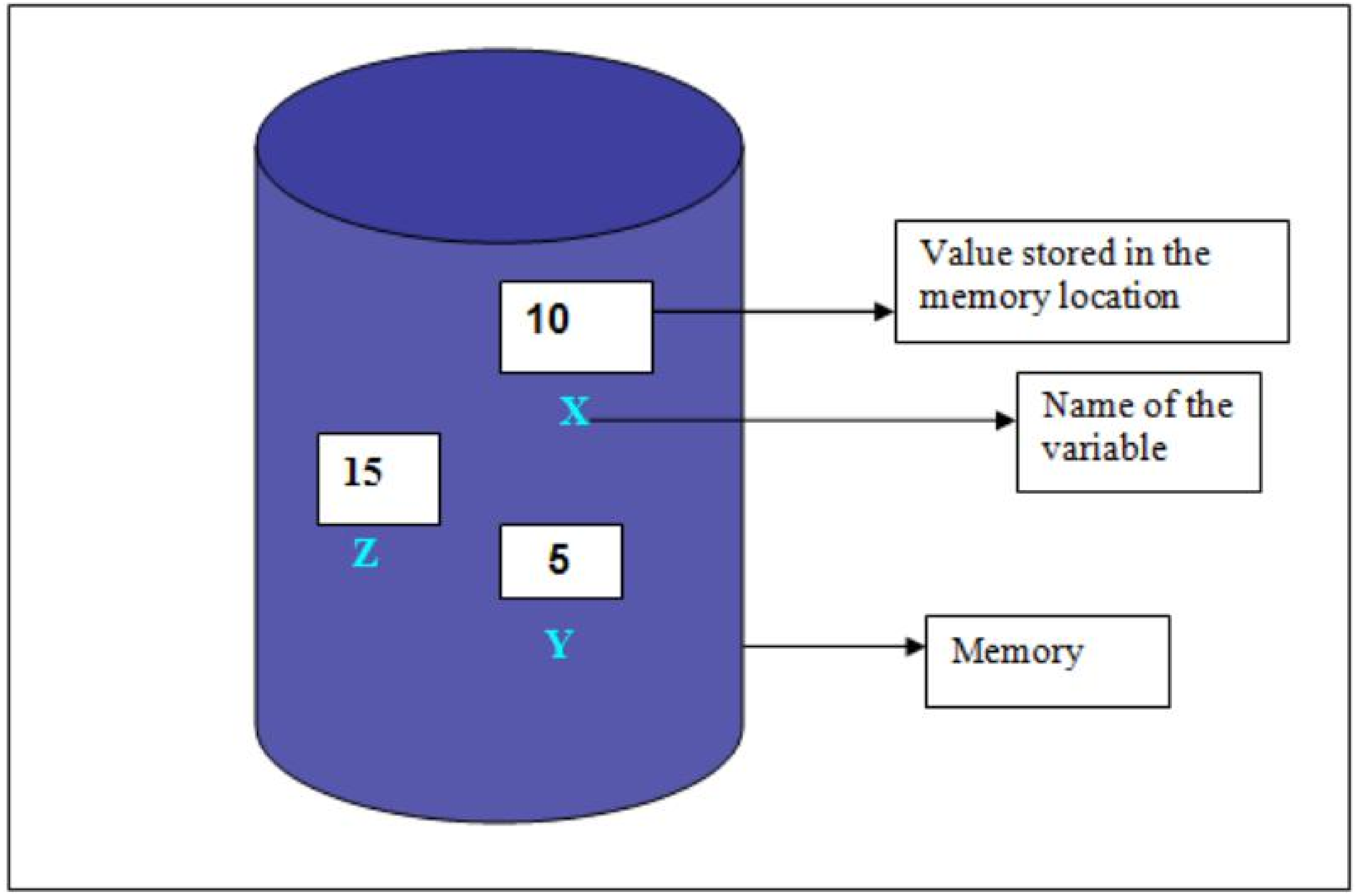


Figure 1: Variables

C programming language

Declaration Variables

DataType_Name Varible_Name ;

- Examples :
- int y;
- long   z ;
- char letter ;
- float f1;
- double salary ;

C programming language

Rules For Declaring Variable

1. The name of the variable contains **letters**, **digits**, and **underscores**.
2. The name of the variable is **case sensitive** (ex **Arr** and **arr** both are different variables).
3. The name of the variable does not contain any whitespace and **special characters** (ex #,$,%,*, etc).
4. All the variable names must begin with a letter of the **alphabet** or an **underscore(_)**.
5. We cannot used **C keyword**(ex float,double,class)as a variable name.

Google Developer Student Clubs

Declaring Variables .

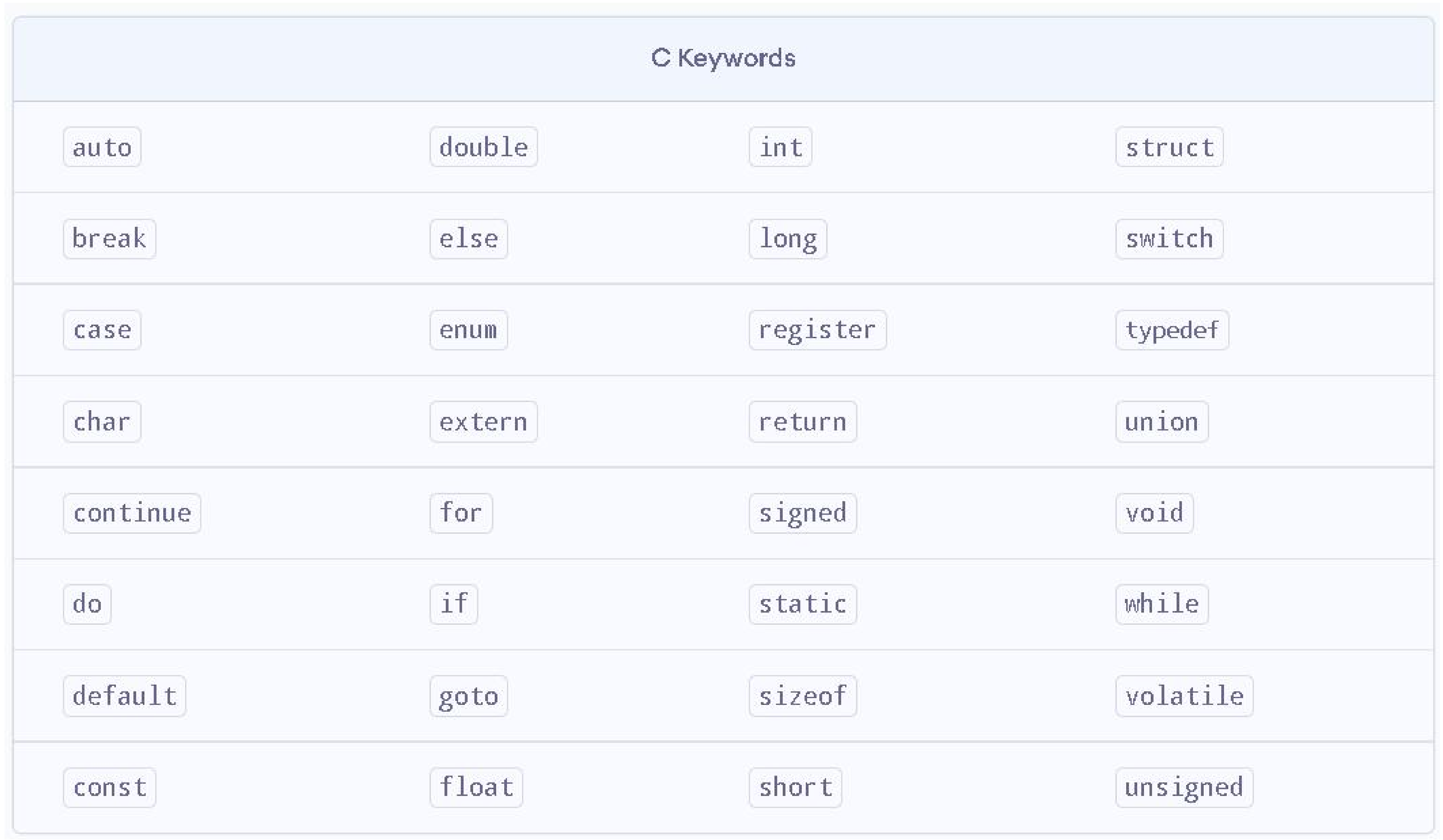
```
#include <stdio.h>

int main() {
    int  1_y;
    long   #z_1 ;
    char letter ;
    bool x_1 ;
    float x#c;
    double x_1_x ;

return 0;

}
```

C programming language



C Keywords			
`auto`	`double`	`int`	`struct`
`break`	`else`	`long`	`switch`
`case`	`enum`	`register`	`typedef`
`char`	`extern`	`return`	`union`
`continue`	`for`	`signed`	`void`
`do`	`if`	`static`	`while`
`default`	`goto`	`sizeof`	`volatile`
`const`	`float`	`short`	`unsigned`

Google Developer Student Clubs

```
#include<stdio.h>
        int main()
{
                int a;
                float b;
                long c;
                double e;
                char f;
                printf("%a\n",a);
                printf("%f\n",b);
                printf("%lld\n",c);
                printf(“%lf\n”\n,e);
                printf(“%c\n”,f);
}
```

Google Developer Student Clubs

C programming language

Initialize Variables

DataType_Name Varible_Name = value ;

OR

DataType_Name Varible_Name ;

Varible_Name = value ;

- int y = 10 ;
- int y;
- y = 10;
- long long z = 92233720368547758 ;
- char letter = ‘x’ ;

```c
#include <stdio.h>
int main()
{
    int x;  // Declaration
    int y = 5; // Declaration and Initialization
    float f; // Declaration
    f = 3.14;  // Initialization
    char c = 'h'; // Declaration and Initialization
}
```

Time to code

Write a program to declare variables :
var1, ***var2***, ***var3***, ***var4***, ***var5***, ***var6***, ***var7***

with data types : ***int*** , ***long long***, ***float***, ***double***, ***char***
And, initialize these variables with these values, respectively: ***5***, ***310000093939***, ***5.34***, ***31.000124***, ***’h’***

```
#include <stdio.h>

int main()

{

   int var1 = 5;

   long var2 = 310000093939;

   float var3 = 5.34;

   double var4 = 31.000124;

   char var5 = 'h';

      printf("%d\n",var1);

      printf("%lld\n",var2);

      printf("%f\n",var3);

      printf("lf\n",var4);

      printf("%c\n",var5);


}
```

C programming language

User Input

In C programming language, scanf is a function that stands for Scan Formatted String. It reads data from stdin (standard input stream i.e. usually keyboard) and then writes the result into the given arguments.

- It accepts character, string, and numeric data from the user using standard input.
- Scanf also uses format specifiers like printf. (%i , %f , %c..,ect)

C programming language

User Input

```
int var;
scanf("%d", &var);
```

called as address of the operator

- &var is the address of var.

While scanning the input, scanf needs to store that input data somewhere.
To store this input data, scanf needs to known the memory location of a variable

```
#include <stdio.h>

int main()

{

    int var1;

    long var2;

    float var3;

    double var4;

    char var5;

        scanf(“%d”,&var1);

        scanf(“%lld”,&var2);

        scanf(“%f”,&var3);

        scanf(“lf”,&var4);

        scanf(“%c”,&var5);


 }
```

Google Developer Student Clubs

A fresh start. A new chapter in life waiting to be written. New questions to be asked, embraced, and loved. Answers to be discovered and then lived in this transformative year of delight and self-discovery.

- Sarah Ban Breathnach

```
function filterStudies({ studies, filterByOrg = false, filte
                        studies.filter(study => {
                                            organizat
```