Programing Methodology in C

Lecture 4 – Keywords, Identifiers, Data Types & Variables

By Elubu Joseph

josebulinga@gmail.com

Agenda

1. C Language Keywords and Identifier
2. Data Types in C
3. Variables in C

C Language Keywords and Identifier

What are Keywords in C?

are preserved words that have special meaning in C language. This is what the compiler know and therefore, these meaning cannot be changed.

For this reason, keywords cannot be used as variable names because that would try to change the existing meaning of the keyword, which is not allowed, Studytonight.com. (n.d.).

The table below gives us a glance on 32 keywords in C language.

32 Keywords in C language

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

Identifiers

are the names given to variables, constants, functions and user-define data. These identifiers are defined against a set of rules.

Rules for defining/Naming an Identifier

1.An Identifier can only have alphanumeric characters(a-z , A-Z , 0-9) and underscore(_).

2.The first character of an identifier can only contain alphabet(a-z , A-Z) or underscore (_).

3.Identifiers are also case sensitive in C. For example **name** and **Name** are two different identifiers in C.

4.Blank spaces are used to separate identifiers

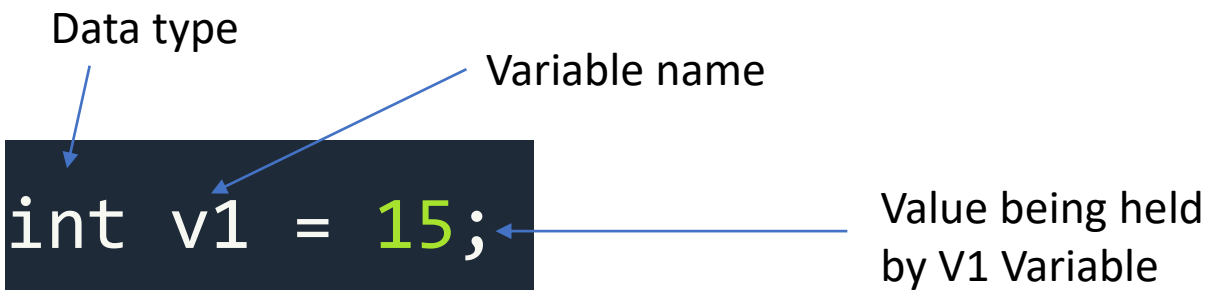
Rules for defining an Identifier+

4.Keywords are not allowed to be used as Identifiers.

5.No special characters, such as semicolon, period, whitespaces, slash or comma are permitted to be used in or as Identifier.

Rules for defining an Identifier++

When we declare a variable or any function in C language program, we must provide its name if it is to be used, which identifies it throughout the program, for example:



Here v1 is the name or identifier for the variable which stores the value “15" in it.

Data types in C Language

A **data type** or simply **type** is a classification of **data** which tells the compiler or interpreter how the **programmer** intends to use the **data**.

Data types specify how we enter data into our programs and what type of data we enter. C language has some predefined set of data types to handle various kinds of data that we can use in our program. These datatypes have different storage capacities.

Categories of Data Types in C Language

There are 5 different categories of data types in the C language, they included:

Type	**Example**
Basic	character, integer, floating-point, double.
Derived	Array, structure, union, etc.
Enumeration	enums
Bool type	true or false
void	Empty value

C Primary/primitive Data types

The 5 basic (primary or primitive) data types in C language are:

1. **Character** - ASCII character set or generally a single alphabet like **(a-z , A-Z).**
2. **Integer** - Used to store whole numbers like **1**, **20**, **100**, **1000,000**, etc.
3. **Floating-point** - Decimal point or real numbers values like **45.9**, **10.5**, etc.
4. **Double** - Very large numeric values which are not allowed in Integer or Floating point type.
5. **Void** - This represent null value. This data type is mostly used when we define functions.

Data Types and their Keywords

There are various keywords to specify these data types, they include:

Datatype	**Keyword**
Character	char
Integer	int
Floating-point	float
Double	double
Void	void

Datatype Modifiers in C language

There are **4 datatype modifiers**, that are used along with the basic data types to help us categorize them further.

E.g. if you say, there is a food, the other person will know that there is food, but you can be more specific and say, there is Chicken or Kimchi

Similarly, there are modifiers in the C language, to **make the primary data types more specific**.

Data type Modifiers in C language+

The following are the modifiers:

1.signed

2.unsigned

3.long

4.short

Data type Modifiers in C language++

signed modifiers are used to represent the **(+ and -) values** while unsigned represent modifiers represent **(only +) values** for any data type. And **long** and **short** modifiers affects the **range of the values** for any datatype.

E.g., **signed int**, **unsigned int**, **short int**, **long int**, etc. are all valid data types in the C language.

Now **let's see modification and the range** for different data types formed as a result of the 5 primary data types.

Data types and Modifiers

Type	Typical Size in Bits	Minimal Range	Format Specifier
char	8	-127 to 127	%c
unsigned char	8	0 to 255	%c
signed char	8	-127 to 127	%c
int	16 or 32	-32,767 to 32,767	%d, %i
unsigned int	16 or 32	0 to 65,535	%u
signed int	16 or 32	Same as int	%d, %i
short int	16	-32,767 to 32,767	%hd
unsigned short int	16	0 to 65,535	%hu

Data types and Modifiers+

Type	Typical Size in Bits	Minimal Range	Format Specifier
signed short int	16	Same as short int	%hd
long int	32	-2,147,483,647 to 2,147,483,647	%ld, %li
long long int	64	$-(2^{63} - 1)$ to $2^{63} - 1$ (Added by C99 standard)	%lld, %lli
signed long int	32	Same as long int	%ld, %li
unsigned long int	32	0 to 4,294,967,295	%lu

Data types and Modifiers++

Type	Typical Size in Bits	Minimal Range	Format Specifier
unsigned long long int	64	2^{64} - 1 (Added by C99 standard)	%llu
float	32	1E-37 to 1E+37 with six digits of precision	%f
double	64	1E-37 to 1E+37 with ten digits of precision	%lf
long double	80	1E-37 to 1E+37 with ten digits of precision	%Lf

Studytonight.com. (n.d.)

So when do use the specifiers

When we want to print the value for any variable with any data type, we have to use a **format specifier** in the printf() statement.

```
#include<stdio.h>
int main() {
        // allowed value up to 65535
        unsigned short int x = 65535;
        printf("the Value of x = %u",x);/*note the use of %u
specifier*/
return 0;
}
```

Memory Location for various data types

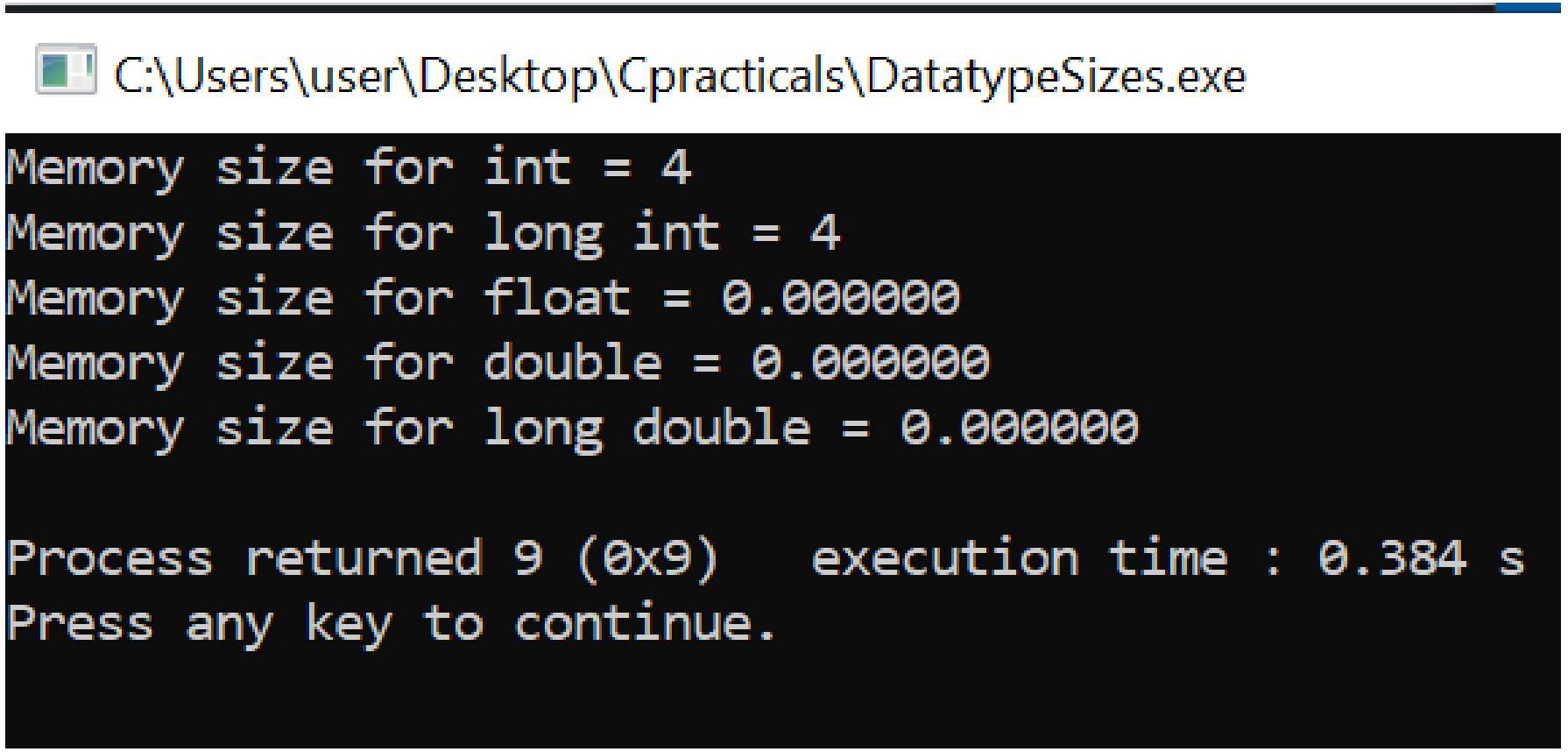
```
#include<stdio.h>
#include<limits.h>
int main(){
    int a;
    long int b;
    float c;
    double d;
    long double e;
```

Linking limits.h library file because we need to use sizeof() function to find ouf memory allocated for each data type

Memory Location for various data types with sizeof() function

```
    printf("Memory size for int = %d\n", sizeof(a));
    printf("Memory size for long int = %ld\n", sizeof(b));
    printf("Memory size for float = %f\n", sizeof(c) );
    printf("Memory size for double = %lf\n", sizeof(d));
    printf("Memory size for long double = %Lf\n", sizeof(e));
return 9;
}
```

Output with the use of **sizeof()** function



Memory Location for Different data types

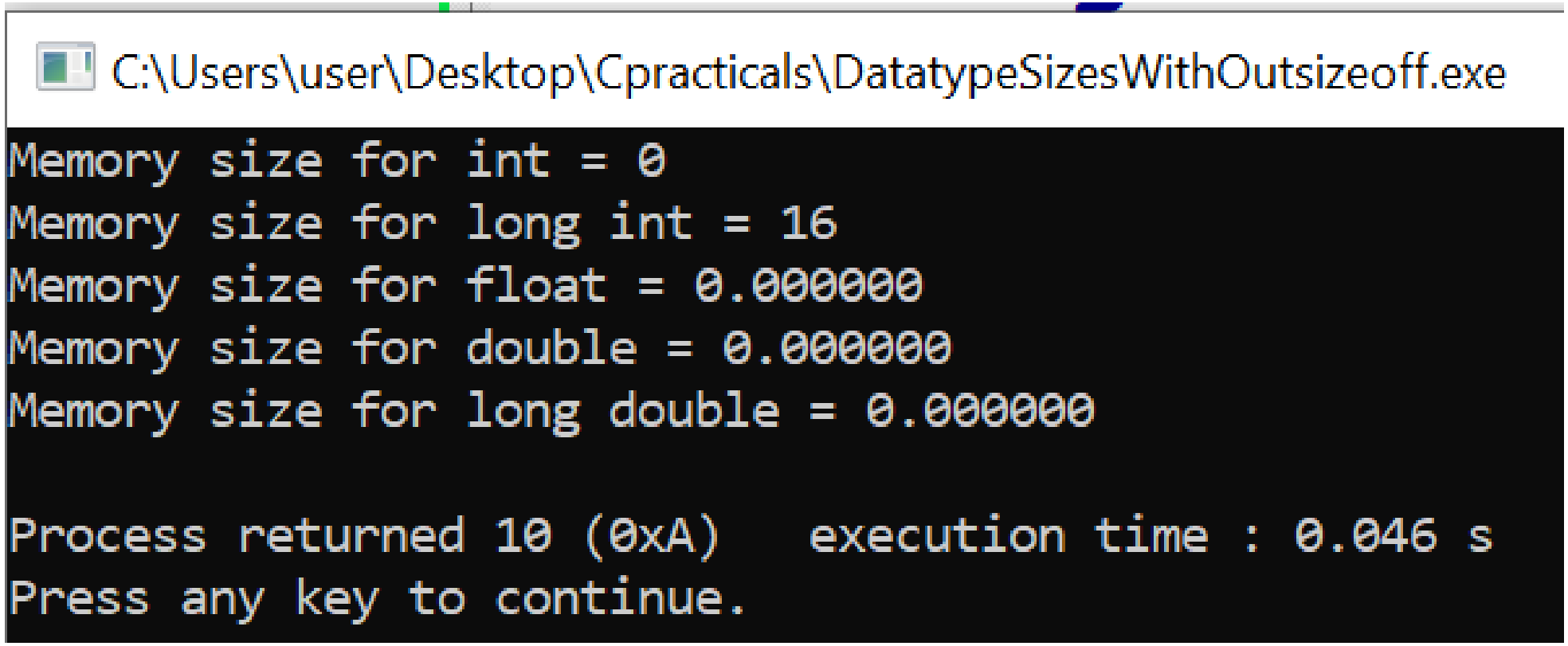
```
#include<stdio.h>
#include<limits.h>
int main(){
    int a;
    long int b;
    float c;
    double d;
    long double e;
```

Linking limits.h library file because we need to use sizeof() function to find ouf memory allocated for each data type

Memory Location for Different data types

```
    printf("Memory size for int = %d\n", sizeof(a));
    printf("Memory size for long int = %ld\n", b);
    printf("Memory size for float = %f\n",c );
    printf("Memory size for double = %lf\n",d);
    printf("Memory size for long double = %Lf\n",e);
return 9;
}
```

Output with the use of without **sizeof()** function



Void type

void type means no value. This is usually used to specify the type of functions which returns nothing.

We will get acquainted to this datatype as we start learning more advanced topics in C language, like functions, pointers etc.

C Derived Datatypes:

While there are 5 primary data types, there are some derived data types too in the C language which are used to store complex data.

- Derived data types are nothing but primary data types, a little twisted or grouped together like an **structure, array**, **union,** and **pointers**.

- These are discussed in detail later.

Variables

Variables in C Language

A variable is a memory holder of data. Or Variable is the name of memory location.

When we want to store any information(data) on our computer, we store it in the computer's memory space.

Instead of remembering the complex address of that memory space where we have stored our data, our operating system provides us with an option to create folders, name them, so that it becomes easier for us to find it and access it.

Variables in C Language

Similarly, in C language, when we want to use some data value in our program, we can store it in a memory space and name the memory space so that it becomes easier to access it.

The naming of an address is known as **variable**. Variable is the name of memory location. Unlike constants, variables are changeable, we can change value of a variable during execution of a program.

A programmer can choose a meaningful variable name. Example : dob, currenYear, height, age, total etc.

Data type Vs Variable

A **variable** in C language must be given a type, which defines what type of data the variable will hold. It can be:

- char: Can hold/store a character in it.
- int: Used to hold an integer.
- float: Used to hold a float value.
- double: Used to hold a double value.
- void represents nothing

Rules for naming Variables

1. Variable name must not start with a digit.
2. Variable name can consist of alphabets, digits and special symbols like underscore.
3. Blank or spaces are not allowed in variable name.
4. Keywords are not allowed as variable name.
5. Upper and lower case names are treated as different, as C is case-sensitive, so it is suggested to NB. keep the variable names in lower case.

Declaring, and Initializing of variables

Declaration of variables must be done before they are used in the program. Variable declaration;-

1.tells the compiler what the variable name is.

2.specifies what type of data the variable will hold.

4.is more like informing the compiler that there exist a variable with following datatype which is used in the program.

5. can be done using the extern or final keyword , outside the main() function.

```
extern int a; extern float b; final double c, d;
```

Initializing a variable

Variable initialization is the act of assigning avalue to agiven variable based on the data type used in declaration.

E.g.

```
Declaration: int a; float b, c;

Initialization: a=20, b=4.5, c=5.1;
```

A variable can be initialized and defined in a single statement, like:

```
int a = 10;
```

Defining a variable

Defining a variable is the act of declaration and initialization of the variable. It is not necessary to declare a variable using extern keyword, if you want to use it in your program. You can directly define a variable inside the main() function and use it.

Let's write a program in which we will use some variables.

```
#include <stdio.h>
	// external Variable declaration(optional)
	extern int a, b;
	extern int c;
int main () {
	/* local variable declaration: */
	int a, b;
	/* actual initialization */
	a = 30;
	b = 14;
	/* using addition operator */
	c = a + b;
	/* display the result */
	printf("Sum is : %d \n", c);
return 0;
}
```

Output:

Sum is : 44

Difference between Variable and Identifier

An identifier is a name given to any variable, function, structure, pointer or any other entity in a programming language. While a variable, is a named memory location to store data which is used in the program.

Difference between Variable and Identifier+

Identifier	**Variable**
Identifier is the name given to a variable, function etc.	While, variable is used to name a memory location which stores data.
An identifier can be a variable, but not all identifiers are variables.	All variable names are identifiers.

Example:
// a variable int kumu;
// or, a function int kumu(){ …
}
	Example:
	// int variable
	int a;
	// float variable float a;

Another great analogy to understand the difference between Identifier and Variable is:

You can think of an Identifier `int x` to be a variable's name, but it can also be a function's name `int x() { }` and still be an identifier.

So what differentiate the two?

`int x;` dose not have the parenthesis while

`int x();` has the parenthesis

Summary

1. C Language Keywords and Identifier

   32 keywords: - int, enum, char, etc.

2. Data Types in C

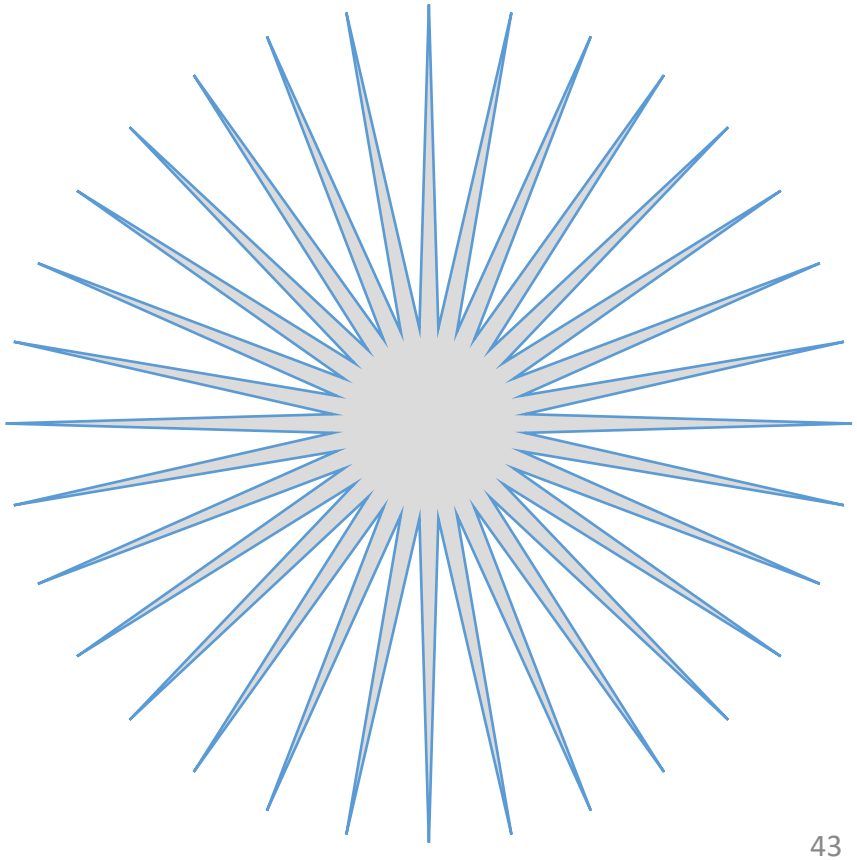
   5 Primary data types and 4 derived data types and data type modifiers.

3. Variables in C:- defined it, gave e.gs, talked about declaration and initialization of variables.

Revision Questions

1. Declare three variable using the following data types; Integer, character and double;
2. What is the difference between variable declaration and variable initialization?
3. Mention any three modifiers you now in C language
4. Write a C program that prints out the memory size allocated to variable **D** declared as type signed long long int point.

Thank you for you attention



References

C keywords and identifiers. Studytonight.com. (n.d.). Retrieved September 7, 2021, from https://www.studytonight.com/c/keywords-and-identifier.php.