Data types

Each program needs a certain kind of data for displaying a meaningful result This certain kind of data are known as a data type.

In C programming, data types are declarations for variables. This determines the type and size of data associated with variables

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.

C language supports 2 different type of data types:

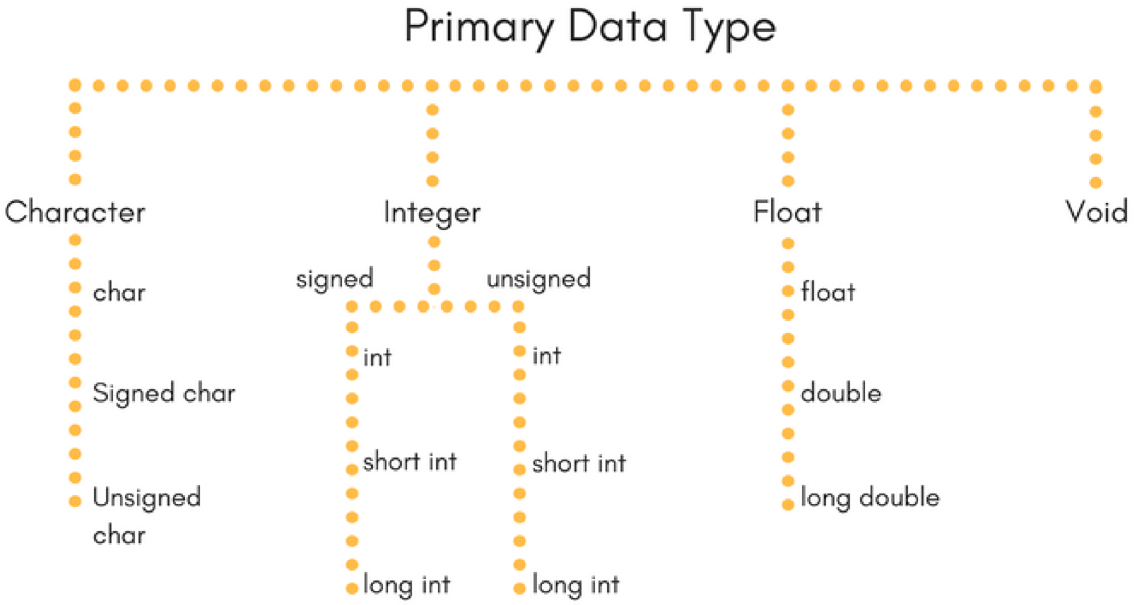
1. **Primary data types**:

   These are fundamental data types in C namely integer(`int`), floating point(`float`), character(`char`) and `void`.

2. **Derived data types**:

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

Data type determines the type of data a variable will hold. If a variable **x** is declared as int. it means **x** can hold only **integer** values. Every variable which is used in the program must be declared as what data-type it is.



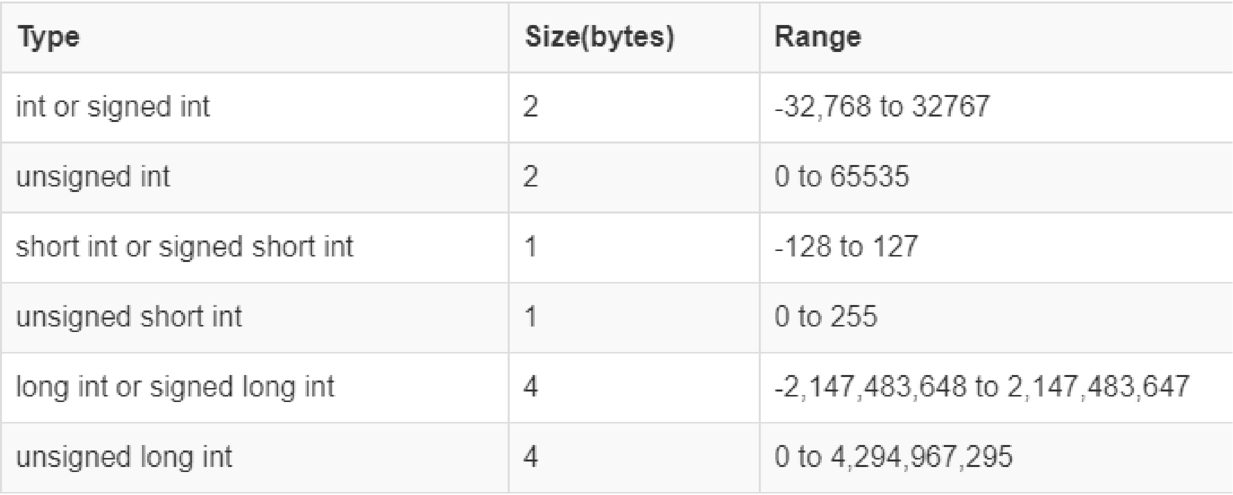
Integer type

Integer is nothing but a whole number. The range for an integer data type varies from machine to machine. The standard range for an integer data type is **-32768 to 32767.**

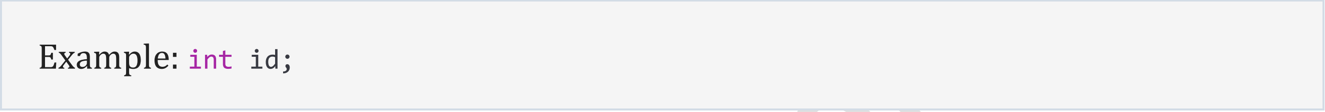
An integer typically is of **2 bytes** which means it consumes a total of **16 bits** in memory. A single integer value takes **2 bytes** of memory. An integer data type is further divided into other data types such as short int, int, and long int.

Each data type differs in range even though it belongs to the integer data type family. The size may not change for each data type of integer family.

The short int is mostly used for storing small numbers, int is used for storing averagely sized integer values, and long int is used for storing large integer values.

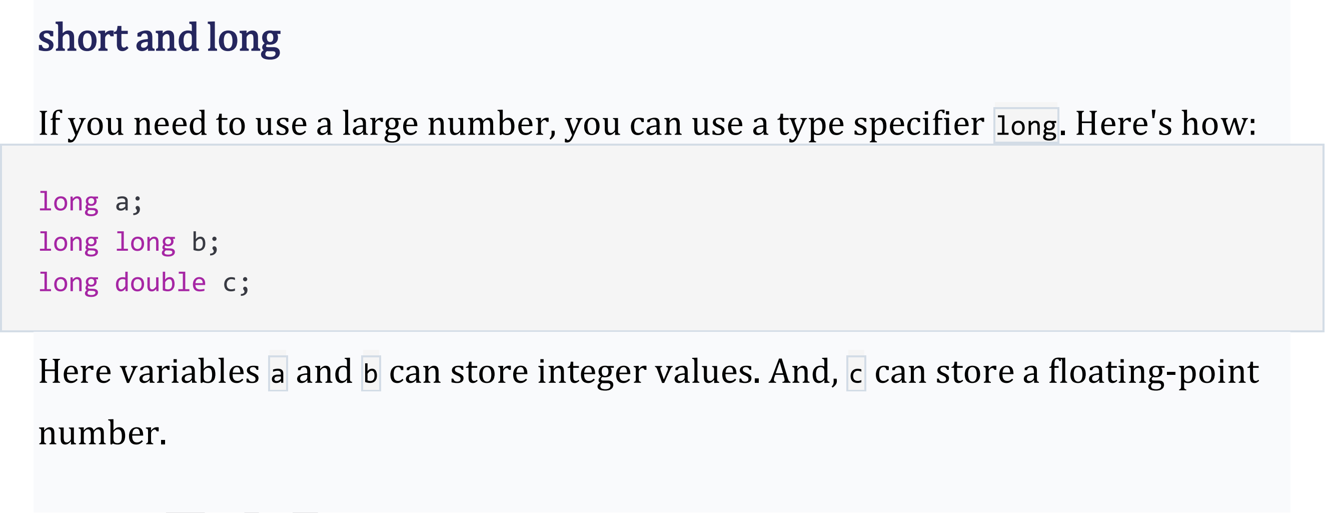


Type	Size(bytes)	Range
int or signed int	2	-32,768 to 32767
unsigned int	2	0 to 65535
short int or signed short int	1	-128 to 127
unsigned short int	1	0 to 255
long int or signed long int	4	-2,147,483,648 to 2,147,483,647
unsigned long int	4	0 to 4,294,967,295



Example: `int id;`

Example: `int id, age;`

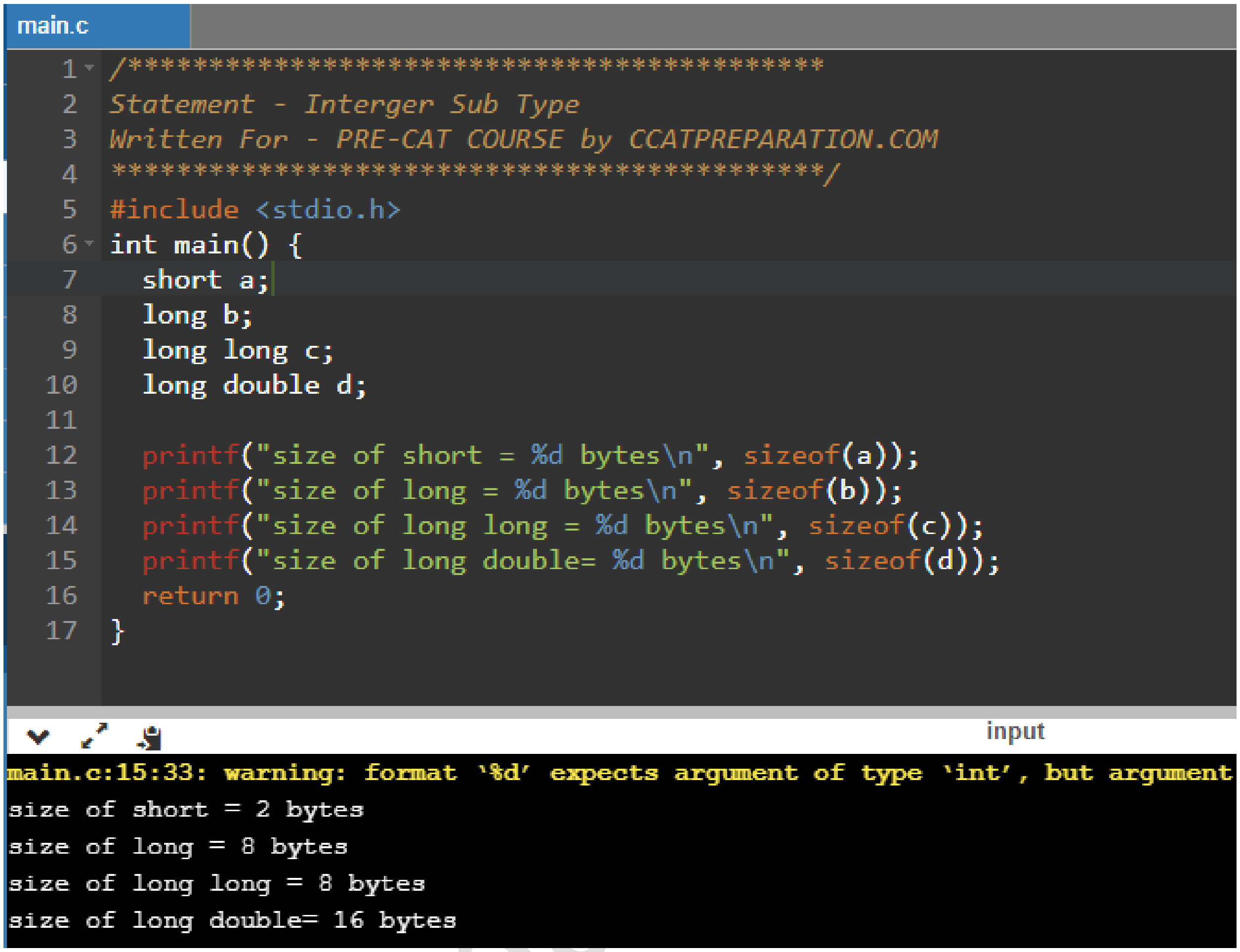


short and long

If you need to use a large number, you can use a type specifier `long`. Here's how:

```
long a;
long long b;
long double c;
```

Here variables `a` and `b` can store integer values. And, `c` can store a floating-point number.



```c
/*******************************************
Statement - Interger Sub Type
Written For - PRE-CAT COURSE by CCATPREPARATION.COM
*******************************************/
#include <stdio.h>
int main() {
  short a;
  long b;
  long long c;
  long double d;

  printf("size of short = %d bytes\n", sizeof(a));
  printf("size of long = %d bytes\n", sizeof(b));
  printf("size of long long = %d bytes\n", sizeof(c));
  printf("size of long double= %d bytes\n", sizeof(d));
  return 0;
}
```

```
main.c:15:33: warning: format ‘%d’ expects argument of type ‘int’, but argument
size of short = 2 bytes
size of long = 8 bytes
size of long long = 8 bytes
size of long double= 16 bytes
```

signed and unsigned

In C, `signed` and `unsigned` are type modifiers. You can alter the data storage of a data type by using them. For example,

```c
unsigned int x;
int y;
```

Here, the variable `x` can hold only zero and positive values because we have used the `unsigned` modifier.

Considering the size of `int` is 4 bytes, variable `y` can hold values from -2^{31} to $2^{31}-1$, whereas variable `x` can hold values from `0` to $2^{32}-1$.

Summary

1. Integer data type allows a variable to store numeric values.
2. "int" keyword is used to refer integer data type.
3. The storage size of int data type is 2 or 4 or 8 bytes.
4. It varies depend upon the processor in the CPU that we use. If we are using 16-bit processor, 2 bytes (16 bit) of memory will be allocated for int data type.
5. Likewise, 4 bytes (32 bit) of memory for 32-bit processor and 8 bytes (64 bit) of memory for 64-bit processor is allocated for int datatype.
6. int (2 byte) can store values from -32,768 to +32,767
7. int (4 byte) can store values from -2,147,483,648 to +2,147,483,647.
8. If you want to use the integer value that crosses the above limit, you can go for "long int" and "long long int" for which the limits are very high.

Note:

1. We can't store decimal values using int data type.
2. If we use int data type to store decimal values, decimal values will be truncated and we will get only whole number.
3. In this case, float data type can be used to store decimal values in a variable.

Floating point type

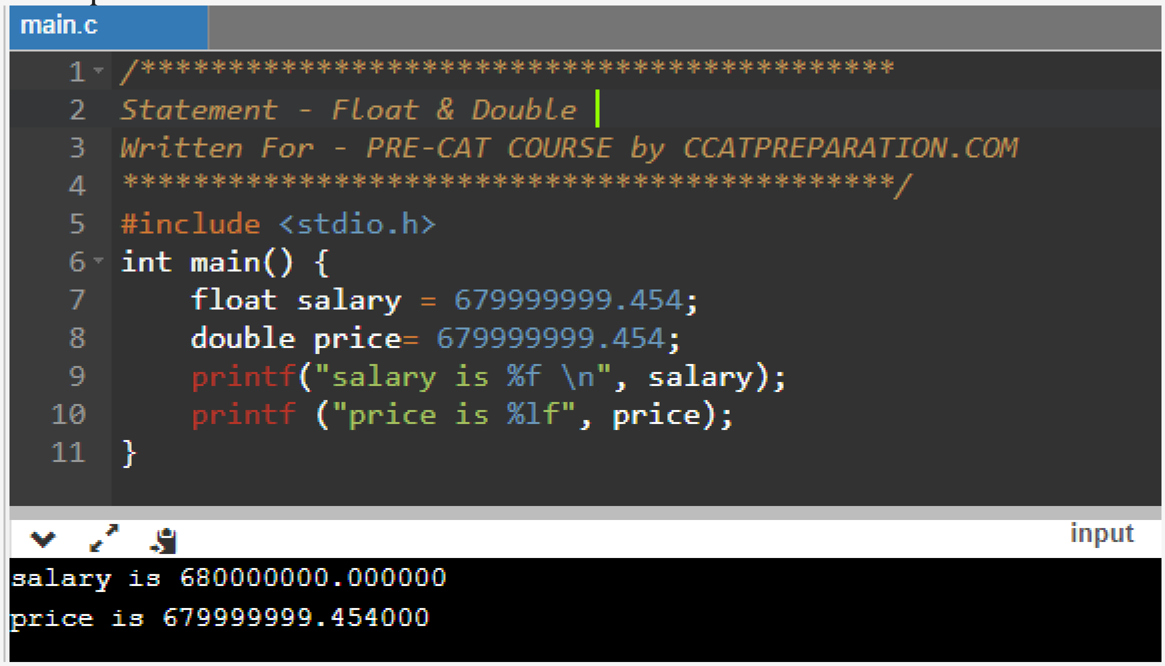
Like integers, in 'C' program we can also make use of floating-point data types. The 'float' keyword is used to represent the floating-point data type. It can hold a floating-point value which means a number is having a fraction and a decimal part. A floating-point value is a real number that contains a decimal point. Integer data type doesn't store the decimal part hence we can use floats to store decimal part of a value.

Generally, a float can hold up to 6 precision values. If the float is not sufficient, then we can make use of other data types that can hold large floating-point values. The data type double and long double are used to store real numbers with precision up to 14 and 80 bits respectively.

While using a floating-point number a keyword float/double/long double must be placed before an identifier.

Type	Size(bytes)	Range
Float	4	3.4E-38 to 3.4E+38
double	8	1.7E-308 to 1.7E+308
long double	10	3.4E-4932 to 1.1E+4932

Example:



```c
/*******************************************
Statement - Float & Double
Written For - PRE-CAT COURSE by CCATPREPARATION.COM
*******************************************/
#include <stdio.h>
int main() {
    float salary = 679999999.454;
    double price= 679999999.454;
    printf("salary is %f \n", salary);
    printf ("price is %lf", price);
}
```

```
salary is 680000000.000000
price is 679999999.454000
```

In C, floating-point numbers can also be represented in exponential. For example,

```c
float Factor = 22.442e2;
```

What's the difference between `float` and `double`?

The size of `float` (single precision float data type) is 4 bytes. And the size of `double` (double precision float data type) is 8 bytes.

Summary :

1. Float data type allows a variable to store decimal values.
2. Storage size of float data type is 4. This also varies depend upon the processor in the CPU as “int” data type.
3. We can use up-to 6 digits after decimal using float data type.
4. For example, 10.456789 can be stored in a variable using float data type.
5. Double data type is also same as float data type which allows up-to 10 digits after decimal.
6. The range for double datatype is from 1E–37 to 1E+37.
7. The ‘signed’ or ‘unsigned’ are not applied on float type.
8. The ‘signed’ or ‘unsigned’ are not applied on double type.
9. The min-values given for ‘float’ and ‘double’ are the minimum possible (the smallest) positive value.

Character type

Character data types are used to store a single character value enclosed in single quotes.

A character data type takes up-to 1 byte of memory space.

Type	Size(bytes)	Range
char or signed char	1	-128 to 127
unsigned char	1	0 to 255

Example:

```
char test = 'h';
```

The 256 ASCII characters (numbered from 0 to 255) can be represented by this type. In ASCII ‘A’ is 65, ‘B’ is 66, ‘C’ is 67, ‘D’ is 68, ‘Z’ is 90, ‘ a’ is 97, ‘b’ is 98, ‘z’ is 122, ‘0’ is 48, ‘1’ is 49, ‘9’ is 57, ‘ ‘ is 32, ‘\n’ is 10, ‘\t’ is 9.

char a=55;

char b=’A’; (is deemed to be equal to) char b=65;

char c=’9’; (is different from) char c=9;

char d=’+’;

Summary Character Type

1. Character data type allows a variable to store only one character.
2. Storage size of character data type is 1. We can store only one character using character data type.
3. “char” keyword is used to refer character data type.
4. For example, ‘A’ can be stored using char datatype. You can’t store more than one character using char data type.
5. Please refer C – Strings topic to know how to store more than one characters in a variable.

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.

Example,

```
void displayData() // Check in Function Notes
```

1. Void is an empty data type that has no value.
2. This can be used in functions and pointers.
3. Please visit “C – Function” topic to know how to use void data type in function with simple call by value and call by reference example programs.

Derived Data Types

Data types that are derived from fundamental data types are derived types. For example: arrays, pointers, function types, structures, etc.

We will learn about these derived data types in later tutorials.

Data Types With Format Specifier

Type	Size (bytes)	Format Specifier
`int`	at least 2, usually 4	`%d`, `%i`
`char`	1	`%c`
`float`	4	`%f`
`double`	8	`%lf`
`short int`	2 usually	`%hd`
`unsigned int`	at least 2, usually 4	`%u`
`long int`	at least 4, usually 8	`%ld`, `%li`
`long long int`	at least 8	`%lld`, `%lli`
`unsigned long int`	at least 4	`%lu`
`unsigned long long int`	at least 8	`%llu`
`signed char`	1	`%c`

Type	Size (bytes)	Format Specifier
`unsigned char`	1	`%c`
`long double`	at least 10, usually 12 or 16	`%Lf`