

## Variables

In programming, a variable is a container (storage area) to hold data. To indicate the storage area, each variable should be given a unique name that is also called (identifier). Variable names are just the symbolic representation of a memory location. Variables can be change value during the execution of program.

### Rules of variable name declarations

The name you assign to a **variable** is also called **identifier**. Some rules of variable declaration are following.

- 1) Variable name must be start alphabet letters or underscore “\_” tart a like salary, name, \_Number etc.
- 2) Identifier/Variable may contain uppercase and lowercase Letters, Numbers, or underscore.
- 3) The if first character “\_”underscore after must be a letter.
- 4) Identifier/Variable cannot contain special character and spaces.
- 5) Identifier/Variable are case sensitive.
- 6) Identifier/Variable cannot contain reserve words or keywords of C/C++ language

**Syntax:** `datatype variable_name ;`

### Convention Of variable:

**DataType** variableName = Content or data ;

**char** Father\_Name[20] = "Muhammad Ali";

**int** ConvanceAllownace = 6500;

**int** \_Salary = 25000;

**char** gr='A';

### Constants/Literals

A constant is a value or an identifier whose value cannot be altered in a program. For example: 1, 2.5, "C programming is easy" etc. As mentioned, an identifier also can be defined as a constant. Constant define must be write starting of C/ C++ program

**Syntax:** `const datatype variable_name = Value ;`

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{
```

```
    const int PI=3.142;
```

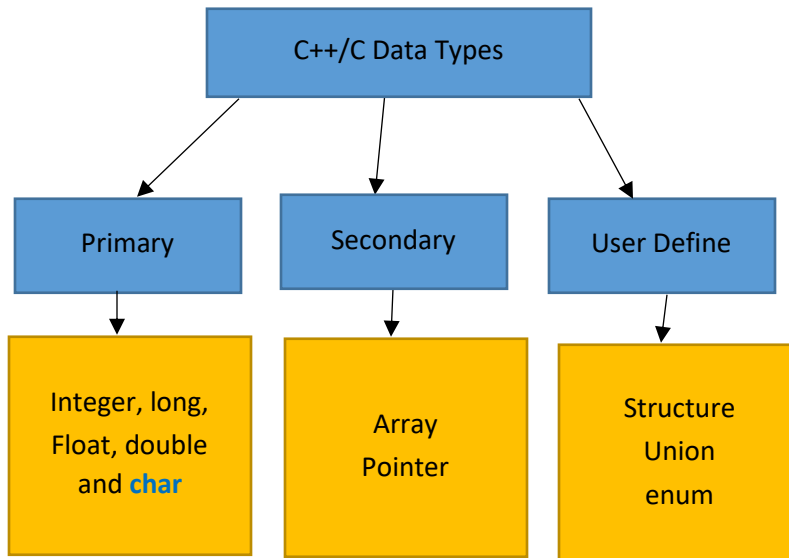
```
    printf("value of PI is %d \n",PI);
```

```
}
```

**Format Specifiers:** It is usual variable type display, specific for each data type like **integer** for **%d** , **character** for **%c** , **long integer** for **%ld** and **float** for **%f** and so on.

**C++ Data Types:**

C++ has a concept of 'data types' which are used to define a variable before its use. The definition of a variable will assign storage for the variable and define the type of data that will be held in the location. There are many data types in C++ / C.



Data types store specified memory size and data values range. There are following tables.

Data Type	Memory (In Bytes)	Format Specifiers	Range
<b>int</b>	2	%d	-32,768 to 32,767
short <b>int</b>	2	%d	-32,768 to 32,767
unsigned <b>int</b>	2	%u	0 to 65535
long <b>int</b>	4	%ld	-2,147,483,648 to 2,147,483,647
unsigned long <b>int</b>	4	%lu	0 to 4,294,967,295
<b>char</b>	1	%c	-128 to 127
Unsigned <b>char</b>	1	%c	0 to 255
<b>float</b>	4	%f	+/- 3.4e +/- 38 (~7 digits)
double	8	%lf	+/- 1.7e +/- 308 (~15 digits)
long double	12	%Lf	-

## Numeric

This data type is based on digits it has further divided into three classes.

- 1) Integer
- 2) Float
- 3) Double

1. **Integer:** This data type contains numeric data up to 2 bytes .Format specifier used for integer data type is %d or %i and it is denoted by “**int**”.
2. **Float:** This data type contains floating point data. It reserves 4 bytes in memory, Format specifier used for **float** data type is %f and it is denoted by “**float**”.
3. **Double:** This data type contains long numeric data. It reserves 8 byte in memory, Format specifier used for double data type is %ld or %lf and it is denoted by “double”.

## Alphabetic /Non Numeric

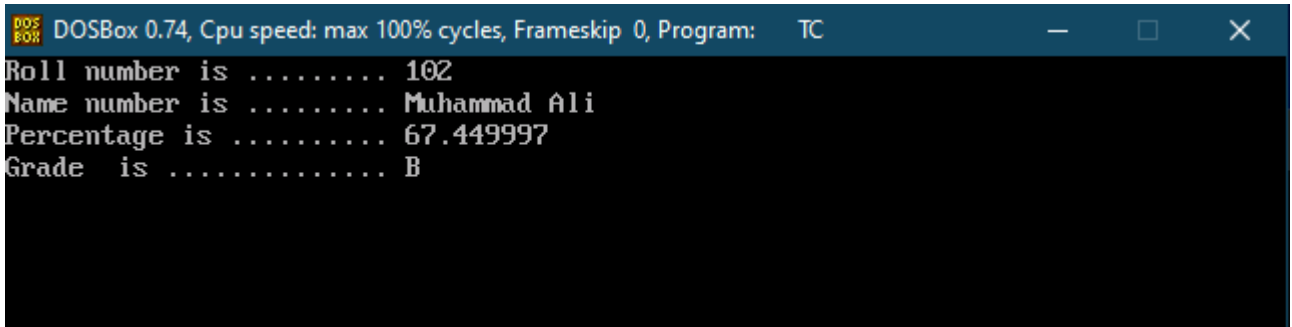
This data type is based on alphabets and alphanumeric data that contains alphabet(s), special character and/or digits. All of these, uses the common data type “**char**”.

1. **char:** This data type contains alphabetic data. It reserves 1 byte/character in memory, Format specifier used for double data type is %c and it is denoted by “**char**”.
2. **string:** Characters data type is also used for multiple characters strings, format specifier use for multiple characters is %s.

### Example

```
#include <conio.h>
#include <stdio.h>
void main()
{
    int Roll=102;
    float Per=67.45;
    char Grade='B';
    char Name[20]="Muhammad Ali";

    clrscr();
    printf("Roll number is ..... %d \n",Roll);
    printf("Name number is ..... %s \n",Name);
    printf("Percentage is ..... %f \n",Per);
    printf("Grade is ..... %c \n",Grade);
}
```



```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
Roll number is ..... 102
Name number is ..... Muhammad Ali
Percentage is ..... 67.449997
Grade is ..... B
```

**Strongly/Static typed Programming**

Strongly or Static typed programming language is in every variable must be declared with a data Type and data type cannot be change after declaration. Java and C++ are strongly or Static typed Programming languages.

**Loosely/Dynamic typed Programming**

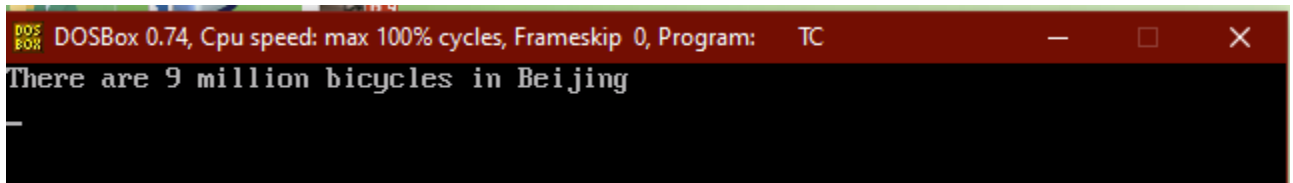
A loosely typed Programming is a programming language that does not require a variable to be defined. You can declare a variable, but it doesn't require you to classify the type of variable. Loose typing is also known as dynamic typing because the data type for a variable can be change after it has been declared. PHP and Perl is a loosely typed programming language.

**Formatting data output with *printf()* function:**

The *printf()* function outputs message with specified format. *printf()* takes a string argument called a format string, usually followed by one or more additional arguments containing the string or strings to format. It then outputs the result.

```
void main()
{
    int number=9;
    char str[]="Beijing";
    printf("There are %d million bicycles in %s \n",number, str);
}
```

The arg1, arg2, ++ parameters will be inserted at percent (%) signs in the main string. This function works "step-by-step". At the first % sign, arg1 is inserted, at the second % sign, arg2 is inserted, etc. one more example of all formatted specified.



```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
There are 9 million bicycles in Beijing
_
```

```

#include <conio.h>
#include <stdio.h>

void main()
{
    short varshort= 32767;
    int varint = 32767;
    long int varlong=123456789;
    float varfloat=123456789.6783;
    double vardouble = -123456789.4567;
    char varchar= 65; // The ASCII Character 65 is A
    char varstr[]="This is a string text ";

    clrscr();
    printf("1) %%d = %d \n",varshort); // short integer number
    printf("2) %%d = %d \n",varint); // Signed decimal number
    printf("3) %%ld = %ld \n",varlong); // Signed long integer number
    printf("4) %%f = %f \n",varfloat); // Signed decimal number
    printf("5) %%lf = %lf \n",vardouble); // Signed double decimal number
    printf("6) %%c = %c \n",varchar); // single character
    printf("7) %%d = %s \n",varstr); // Signed decimal number
    printf("8) %%e = %e \n",varlong); // Scientific notation (lowercase)
    printf("9) %%E = %E \n",varint); // Scientific notation (uppercase)
    printf("10)%%u = %u \n",varint); // Unsigned integer number (positive)
    printf("11)%%F = %F \n",varfloat); // Floating-point number (not local settings aware)
    printf("12)%%g = %g \n",varint); // Shorter of %e and %f
    printf("13)%%G = %G \n",varint); // Shorter of %E and %f
    printf("14)%%o = %o \n",varchar); // Octal number
    printf("15)%%x = %x \n",varchar); // Hexadecimal number (lowercase)
    printf("16)%%X = %X \n",varchar); // Hexadecimal number (uppercase)
    getch();
}

```

```

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
1) %d = 32767
2) %d = -32767
3) %ld = 123456789
4) %f = 12345.672852
5) %lf = 123456789.456700
6) %c = A
7) %d = This is a string text
8) %e = 1.234567e+04
9) %E = 8.882371E+247
10)%u = 32769
11)%g = 8.88237e+247
12)%G = 8.88237E+247
13)%o = 101
14)%x = 41
15)%X = 41

```

Exercise

Theory Questions

- 1) Define five primary data in C++/C Language
- 2) What is format specifier and describe at least 5 format specifier of C Language.
- 3) What difference between variables and constant?
- 4) Explain the purpose of the NULL data type.
- 5) What do you mean by strongly and loosely typed programming?

Practical Questions

- 1) Write a simple program to print your G.R # using variable data type "int".
- 2) Write a simple program of the following output using by *printf* statements.

```
Employee code is 001
Employee Name is ABC
Employee Salary is 25000
Working Hours are 7.5
```

- 3) Repeat the above program using Variables. (Hint: Use 4 Variables respectively for each data.)
- 4) Write down the code of following output, use suitable character or constant(s) if necessary

```
*****          ****   ****   ***
*****   ***   ****   ****   ***
*****   ***   ****   ****   ***
          ***   ****   ****   ***
          *****
          *****
```

- 5) How do you declare a constant in C/C++ with any suitable example?

Objective & MCQ's

- 1) Positive and negative number and 0 with no decimal places belong to which data type.
  - a) Double
  - b) Float
  - c) String
  - d) Integer
  
- 2) Which of the following is a valid variable name?
  - a) TotalSalary ;
  - b) \$Total Salary
  - c) Total Salary;
  - d) Total-Salary;

- 3) How many decimal places does an integer store
  - a) One decimal
  - b) Two decimal
  - c) Three decimal
  - d) Integer does not store decimal places.
  
- 4) Variable name can be starting which symbol.
  - a) \_ (under score)
  - b) @
  - c) \$
  - d) %
  
- 5) String constant value
  - a) Must be enclosed in double quotes
  - b) Must be enclosed in commas
  - c) Must be enclosed in round parentheses
  - d) Must be square brackets.
  
- 6) We use the \_\_\_\_\_ keyword to create constant.
  - a) define
  - b) final
  - c) redefine
  - d) const
  
- 7) A strongly typed programming language \_\_\_\_\_.
  - a) Does not required data typed of a variables to be declared.
  - b) Requires data types of variables to be declared
  - c) Does not have variable
  - d) Does not have different data types.
  
- 8) A constant variable by convention, constant identifiers are \_\_\_\_\_.
  - a) Lower case
  - b) Camel case
  - c) Upper case
  - d) Normal case
  
- 9) Integer value range is \_\_\_\_
  - a) -32,768 to +32,767
  - b) 0 to 65535
  - c) -127 to -128
  - d) -2,147,483,648 to 2,147,483,647