

# Computer Programming - Variables

Variables are the names you give to computer memory locations which are used to store values in a computer program.

For example, assume you want to store two values 10 and 20 in your program and at a later stage, you want to use these two values. Let's see how you will do it. Here are the following three simple steps –

- Create variables with appropriate names.
- Store your values in those two variables.
- Retrieve and use the stored values from the variables.

## Creating variables

Creating variables is also called **declaring variables** in C programming. Different programming languages have different ways of creating variables inside a program. For example, C programming has the following simple way of creating variables –

```
#include <stdio.h>

int main() {
    int a;
    int b;
}
```

The above program creates two variables to reserve two memory locations with names **a** and **b**. We created these variables using **int** keyword to specify variable **data type** which means we want to store integer values in these two variables. Similarly, you can create variables to store **long**, **float**, **char** or any other data type. For example –

```
/* variable to store long value */
long a;
```

```
/* variable to store float value */  
float b;
```

You can create variables of similar type by putting them in a single line but separated by comma as follows –

```
#include <stdio.h>  
  
int main() {  
    int a, b;  
}
```

Listed below are the key points about variables that you need to keep in mind –

- A variable name can hold a single type of value. For example, if variable **a** has been defined **int** type, then it can store only integer.
- C programming language requires a variable creation, i.e., declaration before its usage in your program. You cannot use a variable name in your program without creating it, though programming language like Python allows you to use a variable name without creating it.
- You can use a variable name only once inside your program. For example, if a variable **a** has been defined to store an integer value, then you cannot define **a** again to store any other type of value.
- There are programming languages like Python, PHP, Perl, etc., which do not want you to specify data type at the time of creating variables. So you can store integer, float, or long without specifying their data type.
- You can give any name to a variable like **age**, **sex**, **salary**, **year1990** or anything else you like to give, but most of the programming languages allow to use only limited characters in their variables names. For now, we will suggest to use only **a....z**, **A....Z**, **0....9** in your variable names and start their names using alphabets only instead of digits.
- Almost none of the programming languages allow to start their variable names with a digit, so **1990year** will not be a valid variable name whereas **year1990** or **ye1990ar** are valid variable names.

Every programming language provides more rules related to variables and you will learn them when you will go in further detail of that programming language.

## Store Values in Variables

You have seen how we created variables in the previous section. Now, let's store some values in those variables –

```
#include <stdio.h>

int main() {
    int a;
    int b;

    a = 10;
    b = 20;
}
```

The above program has two additional statements where we are storing 10 in variable **a** and 20 is being stored in variable **b**. Almost all the programming languages have similar way of storing values in variable where we keep variable name in the left hand side of an equal sign = and whatever value we want to store in the variable, we keep that value in the right hand side.

Now, we have completed two steps, first we created two variables and then we stored required values in those variables. Now variable **a** has value 10 and variable **b** has value 20. In other words we can say, when above program is executed, the memory location named **a** will hold 10 and memory location **b** will hold 20.

## Access stored values in variables

If we do not use the stored values in the variables, then there is no point in creating variables and storing values in them. We know that the above program has two variables **a** and **b** and they store the values 10 and 20, respectively. So let's try to print the values stored in these two variables. Following is a C program, which prints the values stored in its variables –

```
#include <stdio.h>

int main() {
    int a;
    int b;

    a = 10;
    b = 20;
```

[Live Demo](#)

```
printf( "Value of a = %d\n", a );  
printf( "Value of b = %d\n", b );  
}
```

When the above program is executed, it produces the following result –

```
Value of a = 10  
Value of b = 20
```

You must have seen **printf()** function in the previous chapter where we had used it to print "Hello, World!". This time, we are using it to print the values of variables. We are making use of **%d**, which will be replaced with the values of the given variable in **printf()** statements. We can print both the values using a single **printf()** statement as follows –

[Live Demo](#)

```
#include <stdio.h>  
  
int main() {  
    int a;  
    int b;  
  
    a = 10;  
    b = 20;  
  
    printf( "Value of a = %d and value of b = %d\n", a, b );  
}
```

When the above program is executed, it produces the following result –

```
Value of a = 10 and value of b = 20
```

If you want to use **float** variable in C programming, then you will have to use **%f** instead of **%d**, and if you want to print a character value, then you will have to use **%c**. Similarly, different data types can be printed using different % and characters.

## Variables in Java

Following is the equivalent program written in Java programming language. This program will create two variables **a** and **b** and very similar to C programming, it will assign 10 and 20 in these variables and finally print the values of the two variables in two ways –

[Live Demo](#)

```
public class DemoJava {  
    public static void main(String []args) {  
        int a;  
        int b;  
  
        a = 10;  
        b = 20;  
  
        System.out.println("Value of a = " + a);  
        System.out.println("Value of b = " + b);  
        System.out.println("Value of a = " + a + " and value of b = " + b);  
    }  
}
```

When the above program is executed, it produces the following result –

```
Value of a = 10  
Value of b = 20  
Value of a = 10 and value of b = 20
```

## Variables in Python

Following is the equivalent program written in Python. This program will create two variables **a** and **b** and at the same time, assign 10 and 20 in those variables.

Python does not want you to specify the data type at the time of variable creation and there is no need to create variables in advance.

[Live Demo](#)

```
a = 10
b = 20

print "Value of a = ", a
print "Value of b = ", b
print "Value of a = ", a, " and value of b = ", b
```

When the above program is executed, it produces the following result –

```
Value of a = 10
Value of b = 20
Value of a = 10 and value of b = 20
```

You can use the following syntax in C and Java programming to declare variables and assign values at the same time –

```
#include <stdio.h>

int main() {
    int a = 10;
    int b = 20;

    printf( "Value of a = %d and value of b = %d\n", a, b );
}
```

[Live Demo](#)

When the above program is executed, it produces the following result –

```
Value of a = 10 and value of b = 20
```