Variables and Assignments

By: Mir Faraz Ali

programming Fundamentals

Variables

Variables are containers for storing data values.

In C++, there are different types of variables (defined with different keywords).

For example:

- **int** - stores integers (whole numbers), without decimals, such as 123 or -123
- **double** - stores floating point numbers, with decimals, such as 19.99 or -19.99
- **char** - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes
- **string** - stores text, such as "Hello World". String values are surrounded by double quotes
- **bool** - stores values with two states: true or false

Variable Declarations & Identifiers

• Identifiers:

- All C++ variables must be identified with unique names.
- These unique names are called identifiers.
- Identifiers can be short names (like x and y) or more descriptive names (age, sum, totalVolume).

• Declaring (Creating) Variables

Syntax:
Datatype variableName = value;

Where Datatype is one of C++ types (such as int), and variableName is the name of the variable (such as x or myName). The equal sign is used to assign values to the variable.

Examples:

- Create a variable called myNum of type int and assign it the value 15:
- int myNum = 15;
- cout << myNum;

- You can also declare a variable without assigning the value, and assign the value later:
- int myNum;
- myNum = 15;
- cout << myNum;

Declare Many Variables:

- To declare more than one variable of the same type, use a comma-separated list:

demo.cpp

```cpp
#include<iostream>
using namespace std;
int main(){
    int x = 5, y = 6, z = 50;
    cout<<x <<endl;
    cout<<y <<endl;
    cout<<z <<endl;

    return 0;
}
```

```
C:\Users\LENOVO\Documents
5
6
50

--------------------------------
Process exited after 0.09071 seconds with return value 0
Press any key to continue . . .
```

Untitled1 demo.cpp

```cpp
#include<iostream>
using namespace std;
int main(){
    int x = 5 ;
    int y = 6 ;
    int z = 50;
    cout<<x <<endl;
    cout<<y <<endl;
    cout<<z <<endl;

    return 0;
}
```

```
C:\Users\LENOVO\Documents
5
6
50

--------------------------------
Process exited after 0.08529 seconds with return value 0
Press any key to continue . . .
```

Real Life Example:

```
// Student data
int studentID = 15;
int studentAge = 23;
float studentFee = 75.25;
char studentGrade = 'B';
// Print variables
cout << "Student ID: " << studentID << "\n";
cout << "Student Age: " << studentAge << "\n";
cout << "Student Fee: " << studentFee << "\n";
cout << "Student Grade: " << studentGrade << "\n";
```

C++ Output:

- The cout object, together with the
- << operator, is used to output values and print text.
- Just remember to surround the text with double quotes (""):

```
#include <iostream>
using namespace std;

int main() {
  cout << "Hello World!";
  return 0;
}
```

- You can add as many cout objects as you want. However, note that it does not insert a new line at the end of the output:

```
#include <iostream>
using namespace std;

int main() {
  cout << "Hello World!";
  cout << "I am learning C++";
  return 0;
}
```

C++ User Input:

- You have already learned that cout is used to output (print) values. Now we will use cin to get user input.
- cin is a predefined variable that reads data from the keyboard with the extraction operator (>>).
- In the following example, the user can input a number, which is stored in the variable x. Then we print the value of x:

Example:

- int x;
- cout << "Type a number: ";
- cin >> x;
- cout << "Your number is: " << x;

End Of Class