Declare Variable

```
variable_type variable_name;
```

Example, declaring a variable called name and used to store user name (text).

```
string name;
```

Note: C++ is case sensitive
Declare variables

C++ allow us to declare multiple variables of same data type in one instruction. Each name separates by a comma.

type variableName1, variableName2;

Ex: 3 variables to store first, middle and last name
string first_name, middle_name, last_name;

If multiple variables have different data types, then one instruction for each data type.

Ex: 4 variables to store first, last name, age, and birth year
string first_name, last_name;
int age, birth_year;
How do we get the data?

- get from the user through console
- hard-coded the data by an equal sign to assign it called *assignment*

```cpp
string name;
cin >> name;
```

cin means get input from console. It follows by a double greater than sign and a variable name. The input will be stored as the data of the variable.

Read in multiple data
```cpp
string fname, lname;
cin >> fname >> lname;
```
How do we get the data?

- get from the user through console
- hard-coded the data by an equal sign to assign it called assignment

```
variable = data;
```

`string name;
name = "Kangmei";

The first time a value is assigned to a variable is called initialization.

```
string name = "Kangmei";

We may assign the same value to multiple variables in one instruction.

```
int age, salary;
age = salary = 0;
```
#include <iostream>
using namespace std;
int main(){
    //declare variables
    string myName, userFName, userLName;
    myName = "Kangmei";
    cout << "Please enter your last name, " << "then first name." << endl;
    cin >> userLName >> userFName;
    /* print out user’s name in format
       first_name last_name with a friendly message, example:
       Hi John Smith, I’m Kangmei.
    */
    cout << "Hi " << userFName << " " << userLName << ", I’m " << myName << "." << endl;
    return 0;
}
## Variable Type

<table>
<thead>
<tr>
<th>Type</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>Store text</td>
</tr>
<tr>
<td>int</td>
<td>Integer – positive, negative whole numbers</td>
</tr>
<tr>
<td>double</td>
<td>Decimals, floating point numbers</td>
</tr>
<tr>
<td>bool</td>
<td>Boolean value – true/false, yes/no</td>
</tr>
<tr>
<td>char</td>
<td>Store a single character – a letter, single digit or a special character. ex: ?, Store as ASCII code</td>
</tr>
</tbody>
</table>

**C++ is case sensitive.**  
**Int ≠ int**
# Variable Name

<table>
<thead>
<tr>
<th>Rules</th>
<th>Variable Name Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>All letters are allowed, Upper case and Lower case</td>
<td>Abc</td>
</tr>
<tr>
<td></td>
<td>xYz</td>
</tr>
<tr>
<td>All integers are allowed, But can’t begin with an integer</td>
<td>Number1</td>
</tr>
<tr>
<td></td>
<td>checkPoint100</td>
</tr>
<tr>
<td>No special character is allowed, except underscore</td>
<td>first_name</td>
</tr>
<tr>
<td></td>
<td>My_Score</td>
</tr>
<tr>
<td>As a convention, variable always start with a lower case letter</td>
<td>pi_value</td>
</tr>
<tr>
<td></td>
<td>bookTitle</td>
</tr>
<tr>
<td>Can’t use C++ keywords</td>
<td></td>
</tr>
<tr>
<td>C++ key words</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td>Asm</td>
<td>auto</td>
</tr>
<tr>
<td>catch</td>
<td>char</td>
</tr>
<tr>
<td>default</td>
<td>delete</td>
</tr>
<tr>
<td>enum</td>
<td>dynamic_cast</td>
</tr>
<tr>
<td>for</td>
<td>union</td>
</tr>
<tr>
<td>goto</td>
<td>if</td>
</tr>
<tr>
<td>mutable</td>
<td>virtual</td>
</tr>
<tr>
<td>private</td>
<td>protected</td>
</tr>
<tr>
<td>reinterpret_cast</td>
<td>return</td>
</tr>
<tr>
<td>static</td>
<td>static_cast</td>
</tr>
<tr>
<td>template</td>
<td>this</td>
</tr>
<tr>
<td>typedef</td>
<td>typeid</td>
</tr>
</tbody>
</table>
Declare Variable Examples

Valid:
string address;
string zipCode;
int age;
int x;
double pi;
bool checkPoint1;
char second_Character;

Not valid:
INT x;
number y;
string firstName;
int 5;
bool yes?;
int string;
string ‘welcome!’;