Identifier, Variable, and Constants

- **Identifier**
  An identifier is the name used for a data object (a variable, or a constant), or for a function, in a C++ program

  - An identifier may start with a letter or underscore, and be followed by zero or more letter (A-Z, a-z), or digits (0-9), or underscore

    Valid identifier: age_of_dog, taxRate, PrintSummary, Y2K,…
    Not valid identifier: age#, 2000taxrate, Dog-and-Cat, C++, main

  - C++ reserved words have unique meanings special to C++, can not be used for other purposes.
  - C++ is case sensitive testScore, testscore, Testscore are all considered different
  - Use meaningful identifiers is a good programming practice: averageScore, numOfStudents, monthlyRent, ComputeSum, PI,
  - Identifier naming conventions
    - Variables: begin with a lowercase letter and capitalize each successive word.
    - Constants: capitalize every letter and use underscores to separate the English words
    - Programmer written functions: will be capitalized in the same way as variable names but they will begin with a capital letter.

- **Variable**: a location in memory, referenced by an identifier, that contains a data value that can be changed.
  - Each variable has a type associated with it indicating what type of data (data type) will be stored in that memory location
  - The type of variable determines the amount of memory space to be allocated for the variable
  - Basic C++ data types:

Data type	**type of data to be stored**
int	positive or negative whole numbers (e.g., age, numberOfTests, numOfRooms …)

Table 2-6 Integer Data Types

Data Type	Typical Size	Typical Range
`short int`	2 bytes	−32,768 to +32,767
`unsigned short int`	2 bytes	0 to +65,535
`int`	4 bytes	−2,147,483,648 to +2,147,483,647
`unsigned int`	4 bytes	0 to 4,294,967,295
`long int`	4 bytes	−2,147,483,648 to +2,147,483,647
`unsigned long int`	4 bytes	0 to 4,294,967,295
`long long int`	8 bytes	−9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
`unsigned long long int`	8 bytes	0 to 18,446,744,073,709,551,615

char an individual character (1 byte)
(enclosed in single quote marks, e.g., 'a', 'W', 'y', 'Y', '\n', '\t', ….)
numeric value of the character from the character set is stored in the memory

CODE:
```
char letter;
letter = 'C';
```

MEMORY:
letter

string a sequence of zero or more characters enclosed in double quotes
(enclosed in double quote marks, e.g., "My name is", "good job!", "end of semester\n", "" …)

string is not a built-in (standard) type
it is provided in the C++ standard library, need to include the C++ standard library header file <string>

float real number or floating number
double (e.g., temperature, testAverage, accountBalance …)
long double

Table 2-8 Floating Point Data Types on PCs

Data Type	Key Word	Description
Single precision	`float`	4 bytes. Numbers between ±3.4E-38 and ±3.4E38
Double precision	`double`	8 bytes. Numbers between ±1.7E-308 and ±1.7E308
Long double precision	`long double*`	8 bytes. Numbers between ±1.7E-308 and ±1.7E308

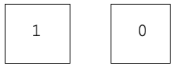
```
float distance;
double mass;

distance = 1.495979E11;
mass = 1.989E30;
```
← *scientific notation*

bool **represent value true (1) or false (0)**

```
bool allDone = true;
bool finished = false;
```

allDone finished



- **Variable declaration**

Basic syntax: ***data-type variable-name;***

Examples:
1. To declare a variable to store the balance of a person's saving account
   float balance;

2. To declare a variable to store the number of employee in a company
   int numberEmployee;

3. To declare a variable to store a user's answer ('y'/'Y' for yes, 'n'/'N' for no) as to whether to continue playing the computer game

```
char     answer;
```

4. To declare a variable to store the name of a customer
```
string    name;
```

5. C++ 11 introduces an alternative way to define variables, using the auto key word and an initialization value. Here is an example:
```
auto amount = 100;          ← int
```
The auto key word tells the compiler to determine the variable's data type from the initialization value.
```
auto interestRate= 12.0;  ← double   (default type for floating value values)
auto stockCode = 'D';      ← char
auto customerNum = 459L;   ← long
```

- **Named constant**
A location in memory, referenced by an identifier, that contains a data value that **cannot** be changed.
  - A named constant can be of any data type used by a variable, e.g., int, float, char, string, …
  - Declare named constants in global variable/function declaration section
  - Declare and assign the value of a named constant

  Basic syntax : ***const data-type constant-name = constant-value ;***

  Examples:
  1. declare a constant to hold the minimum wage of a state
  ```
  const float  MINIMUM_WAGE = 6.5;
  ```
  2. declare a constant to hold the legal age for driving in TN
  ```
  const int LEGAL_AGE = 16;
  ```

- **Literal value:** Any constant value written in a program:

Data type	**literal examples**
int	100, -21, 0, …
float	98.5, 3.1415926, …
char	'a', 'M', '\n', …
string	"Total Cost", "Enter name : ", ….