Array

- A collection of similar data of the same type

Model:
```
datatype_of_array name_of_array[size_of_array]
```

- datatype_of_array: the data type of which stored in the array.
  Ex: int

- name_of_array: name of this array which could be referred to later.
  Ex: grades

- size_of_array: capacity of this array, how many items can be stored in.
  Ex: 10

Example:
```
int grades[10];
string names[10];
```
## Variable Datatype

<table>
<thead>
<tr>
<th><strong>Data Type</strong></th>
<th><strong>Use</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>Store text</td>
</tr>
<tr>
<td>int (primitive/build-in)</td>
<td>Integer – positive, negative whole numbers</td>
</tr>
<tr>
<td>double (primitive)</td>
<td>Decimals, floating point numbers</td>
</tr>
<tr>
<td>bool (primitive)</td>
<td>Boolean value – true/false, yes/no</td>
</tr>
<tr>
<td>char (primitive)</td>
<td>Store a single character – a letter, single digit or a special character. ex: ? , Store as ASCII code</td>
</tr>
<tr>
<td>Other Customer Defined - Datatypes</td>
<td>Better suit for other purposes</td>
</tr>
</tbody>
</table>
Array

When referring by the array name, it’s referring the entire array

Ex:
```c
int grades[10];
grades:
```

Print( grades );
Accessing Array

- only ONE data can be accessed at a time.
- we need to specify the index/position of this element in the array

```c
int grades[10]; ← declaration
grades[0]; grades[1]; grades[2]; ... grades[9];
```

grades[ index ]

index from 0 – capacity - 1
Accessing Array

Ex: int grades[10];

Assign first data to 100
grades[0] = 100;

Output first data to console
cout << grades[0];

Input first data from console
cin >> grades[0];
Use loops to access entire Array

To access all the elements in the array, since the only thing it changes is the index/position. Thus, we can use a loop!

Ex: int grades[10]

Output the data from entire array
for (int idx = 0; idx < 10; idx++) {
    cout << grades[idx] << endl;
}
Initializing Array

Sometime we would like to initialize the array

```
int lookup[5] = { 100, 90, 80, 70, 60};
```

Special case: Initialize all to zero

```
int sum[10] = {0};
{0} is a special code, {1} or others won’t work!
```