C++ for Science and Engineering COSC3000/6000

2018 Spring Semester

Part XIII
Arrays and Class

1 Arrays and Classes

- Arrays can use structures or classes as their base types
  - Example:

```
class WindInfo
{
        public:
        double velocity;
        char direction;
};
:
:
WindInfo data_point[10];
```

1.1 Accessing Members

- When an array's base type is a structure or a class...
  - Use the dot operator to access the members of an indexed variable
  - Example:

```
for (i = 0; i < 10; i++)
{
        cout << "Enter velocity: ";
        cin >> data_point[i].velocity;
}
```

1.2 Arrays as class Members

- A structure can contain an array as a member
  - Example:

```
calss Data
{
        public:
        double time[10];
        int distance;
};
:
:
Data my_best;
```

  - **my_best** contains an array of type **double**

- To access the array elements within a structure
  - Use the dot operator to identify the array within the structure
  - Use the [ ]'s to identify the indexed variable desired
  - Example:
    ```
    my_best.time[i]
    ```
    references the i-th indexed variable of the variable time in the structure **my_best**

1.2.1 Arrays as Class Private Members

- Since private members are not accessible from outside, “mutator” and “accessor” are needed.
  ```
  class Data
  {
  public:
      // mutator for time
      void set_time(int n, int ti);
      // accessor for time
      int get_time(int n);
  private:
      // size of array (note staic const to define a constant)
      static const int size = 10;
      double time[size];
      int distance;
  };

  void Data::set_time(int n, int ti)
  {
      if (n >= 0 && n < size){
          time[n] = ti;
      }
  }
  int Data::get_time(int n)
  {
      if (n >= 0 && n < size){
          return time[n];
      }
      return -1;
  }
  ```
  - Note that the constant value for this class, “size” define as “static”.

- in main,
  ```
  int main(int argc, char *argv[])
  {
      Data my_best;
      my_best.set_time(0,10);
  }
  ```

1.3 Working with constructors

- Here we want to define an arrys of **TimeOfDay** class like:

-
  ```
  TimeOfDay time_point[10];
  ```

- Which constructor does above declaration use?

  - It will use the default constructor.

- We did not implement “mutator” for “TimeOfDay” class. To assign time, we have to do

  ```
      for (int i = 0 ; i < 10 ; i++){
          time_point[i] = TimeOfDay(100 * i);
      }
  ```

1.3.1 Dynamic Array

we can define an arrys of **TimeOfDay** class dynamically:

```
TimeOfDay *time_point = new TimeOfDay[10];
```

- We can free the array by

  ```
  delete [] time_point;
  ```