

1 Going into detail of class definitions

1.1 Summary for classes

Properties of a class :

- Corresponds to a data-structure, defined with several **data fields** ;
- each data field has a **type** and an **identifier** ;
- data fields can be **public** or **private** ;
- a instantiation of a class is called an **object** and is the same as a variable ;
- **member functions** are functions that can be applied to an object and have privileged access to the data fields ;
- member functions are called with either the `.` operator or the `->` operator if we use a pointer to an object ;
- **constructors** are special functions called when creating an instance of the class, they do not return types and have for identifier the same identifier as the class itself ;
- **the destructor** is a special member function called when an object is destructed, is has no return value and has for identifier the class name prefixed by a `~` ;
- we can also define **member operators** ;
- we can define member function out of the class statement by using the `<class name>::<member name>` syntax.

1.2 Example

```
1  #include <iostream>
2
3  class SimpleClass {
4      char *name;
5      int value;
6  public:
7      SimpleClass(char *n, int v) {
8          cout << " " << n << ".SimpleClass("
9              << n << ", " << v << ")\n";
10         name = n; value = v;
11     }
12     ~SimpleClass() {
13         cout << " " << name << ".~SimpleClass()\n";
14     }
15     void changeValue(int v) {
16         cout << " " << name << ".changeValue(" << v << ")\n";
17         value = v;
18     }
19     int readValue() {
20         cout << " " << name << ".readValue()\n";
21         return value;
22     }
23     int copy(SimpleClass &sc) {
24         cout << " " << name << ".copy(" << sc.name << ")\n";
25         value = sc.value;
26     }
27 };
```

```
1 int main(int argc, char **argv) {
2     SimpleClass x("x", 12);
3     SimpleClass y("y", 14);
4     x.copy(y);
5     cout << x.readValue() << '\n';
6     y.changeValue(10);
7     cout << y.readValue() << '\n';
8 }
```

```
1     x.SimpleClass(x, 12)
2     y.SimpleClass(y, 14)
3     x.copy(y)
4     x.readValue()
5 14
6     y.changeValue(10)
7     y.readValue()
8 10
9     y.~SimpleClass()
10    x.~SimpleClass()
```