C++ Objects

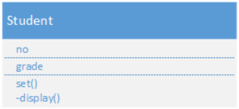
Eden Burton <ronald.burton@senecacollege.ca>
github repository:
(https://github.com/Seneca-OOP244/SCD-Notes)

02/10/2015

Objects

- a instance of a compound type (class/struct)
- a region of memory
- class definition determines…
  - size of memory region
  - how to access various field members
  - which other objects can access its contents
  - `Student s;`
    - allocated in static memory region
  - `Student s = new Student()`
    - allocated in dynamic memory region

Object - Student Example



\\ Student.h, class declaration

```
class Student {
   int no;
   char grade;
public:
   void set(int, const char);
   void display();
}
```

\\ main.cpp, class definition

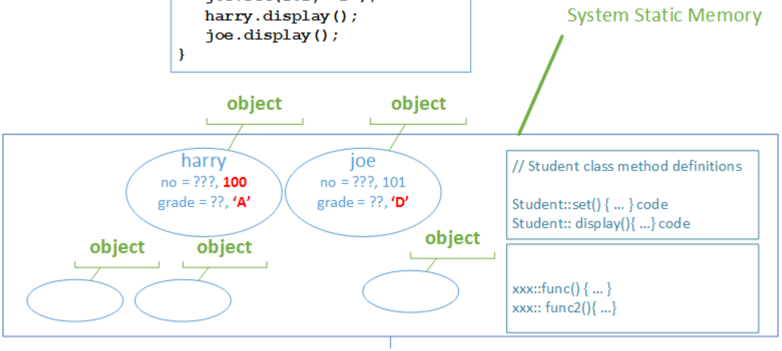
```
#include “Student.h”
using namespace std;

int main() {
   Student harry, joe
   harry.display();
   joe.display();
   harry.set(100, ‘A’);
   joe.set(101, ‘D’);
   harry.display();
   joe.display();
}
```

\\ Student.cpp, class definition

```
#include “Student.h”
using namespace std;

void Student::set(int n, char g) {
   no = n;
   grade = g;
}

void Student::display() {
   cout << no << ‘:’ << grade
}
```

Object - Memory View

```
\\ main.cpp, class definition

#include "Student.h"
using namespace std;

int main() {
   Student harry, joe;
   ...
   harry.display();
   joe.display();
   harry.set(100, 'A');
   joe.set(101, 'D');
   harry.display();
   joe.display();
}
```



```
// Student class method definitions

Student::set() { ... } code
Student:: display(){ ...} code
```

```
xxx::func() { ... }
xxx:: func2(){ ...}
```

Object Privacy

- privacy is implemented at the class level
- a class method can access private members of the “current" object and others of the same type

```
void Student::set(const Student& src) {
    no = src.no;
    strcpy(grade, src.grade);
}
```

Constructors

- special member function that is automatically executed when an object is created
- can have multiple ones.....
- syntax is `ClassName()`
- ensures a safe initial state

Object Construction

1. allocate memory
2. executes constructor logic

Destructors

- special member function that is automatically executed just before an object is destroyed
- syntax is `~ClassName()`
- does clean up of resources it is using
- cannot be overloaded (why?)

Object Destructor

1. executes destructor logic
2. deallocate memory

“This" Object

... a member function executed on a specific object ...

Member Function Parameters

- explicit, interact with client code, passed into function
- implicit, instance members of the current object

```
Student::Student(int n, const char g) {
        no  = n;
        grade  = g);
```

“This" Pointer

- can be used to disambiguate implicit and explicit parameters
- `*this`, refers to object

```
Student Student::set(int no, const char grade) {
     this->no = no;
     this->grade = grade;
     return *this;
 }
```