CSC340: Dynamic Arrays in a Class--''The Big Three''

Main Topics:
1. **The big three: destructor, copy constructor and overloaded assignment operator**
2. **Advanced topic: the move constructor**

Readings:
1. **5th Edition: pages 194--199 (copy constructor and destructor, pages 428—430 (overloaded assignment operator)**

Hui Yang
Computer Science Department
San Francisco State University
http://www.cs.sfsu.edu/~huiyang/

The Big Three

- **Required if a class includes a data member that requires dynamic memory allocation and management such as a dynamic array or a linked list**
  - this class MUST include the "big-three"
- **What are the ''big-three''?**
  - The copy constructor
  - The assignment operator
  - The destructor
- **If you need to define one, you need to define all**

Program Example

- **The Student class**
  - Dynamic data members

    ```
    string* email_list;
    int num_emails;
    double *grade_list;
    int num_grades;
    ```

  - The big-three

    ```
    ~Student(); //destructor
    Student(const Student& ); //copy constructor
    Student operator=(const Student& rhs); //assignment
    ```


Destructor

- **A destructor is a member function that is called automatically when an object of the class goes out of scope**
  - Contains code to delete all dynamic variables created by the object
  - A class has only one destructor with no arguments
  - The name of the destructor
    - Example: `~Student( );`
- **Why destructor?**
  - Dynamic variables do not "go away" unless *delete* is called
  - A user of the Student class could not know that a dynamic array is a member of the class, so could not be expected to call delete when finished with a Student object

Copy Constructors

- **A copy constructor is a constructor with one parameter of the same type as the class**
  - The parameter is a call-by-reference parameter
  - The parameter is usually a constant parameter
  - The constructor creates a complete, independent copy of its argument
- **Syntax**
  - Student Student(const Student&);

The Need For a Copy Constructor

- **This code (assuming no copy constructor) illustrates the need for a copy constructor**
  - void print_Student(Student john_local)
    { ...}

    Student john;
    print_Student(john);
    cout << john << endl; //**memory fault!!!**

Calling a Copy Constructor

- **A copy constructor can be called as any other constructor when declaring an object**
- **The copy constructor is called automatically**
  - When a class object is defined and initialized by an object of the same class
  - When a function returns a value of the class type
  - When an argument of the class type is plugged in for a call-by-value parameter

The Assignment Operator

- **Given these declarations:**

  Student Mary, Lisa;

  the statement

  Mary = Lisa;

  is legal
- **But, there is a problem!**
  - Both Mary and Lisa will point to the same memory location for email_list and grade_list.
  - Violation of information encapsulation!
  - More seriously: the program is buggy and will lead to runtime error!

Overloading =

- **The solution is to overload the assignment operator =**
  - operator = is overloaded as a member function
  - Example: operator = declaration

```
Student operator=(const Student& rhs);
```

- **Demo**

The move constructor

- **Recall: when a function returns a Student object, Student's copy constructor will be automatically called.**
- **Efficiency concern:**
  - The object to be returned has a very short life span
  - Both copy constructor and destructor are called to make a copy of the object and then destroys it.
- **Solution: the move constructor**

```
Student( Student && s);   //move constructor

//Note: the following implementation is incomplete
Student::Student( Student && s):email_list(s.email_list), num_emails(s.num_emails){
    s.email_list = nullptr;  s.num_emails = 0;
}
```

Summary

- If a class contains a dynamic data member, this class must include and implement the Big-3
  - Destructor
  - Copy constructor
  - Assignment operator
- Need to understand the necessity of including big-3
- The move constructor

Quiz

- A program needs to explicitly call a class's destructor to return the allocated memory.
- The assignment operator can only be overloaded as a member function.
- The copy constructor of class Student will be automatically called when a function has a call-by-value parameter whose data type is Student.