CS32 Intro to Computer Science II

Baoxiong Jia & Muthu Palaniappan, DIS 1C Week 2
UCLA Spring 2021

About Us

- TA: Baoxiong Jia
  - Email: baoxiongjia@cs.ucla.edu
  - Office Hours: Tuesday 8:30-10:30am
  - Thursday 8:30-10:30am
  - Discussion 1C: Friday 12:00-13:50pm

- LA: Muthu Palaniappan
  - Email: muthupal@g.ucla.edu
  - Office Hours: Monday 10:30-11:30am
  - Wednesday 10:30-11:30am

Outline

- Constructor, destructor
- Copy constructor
- Assignment operator
- Initializer list
- Order of construction
- Worksheet problems

Constructor & Destructor

- Constructors are called when an object of the class is created
  - Default constructor is a constructor with **no arguments**. Could be explicitly declared, or implicitly by the compiler.
- Destructors are invoked automatically when the object goes out of scope or is explicitly destroyed by a call to delete.
  - Not releasing the memory space for out of scope objects will cause **memory leak**.

```
30  String::String(const char* value)
31  {
32      if (value == nullptr)
33          value = "";
34      m_size = strlen(value);
35      m_text = new char[m_size+1];
36      strcpy(m_text, value);
37  }
```

```
46  String::~String()
47  {
48      delete [] m_text;
49  }
```

Copy constructor

- Initializes an object of your class using another object of the same class
  - When your class doesn't have a copy constructor, C++ will implicitly make one for you, simply copy all the references over. (**shallow copy**)
  - The default copy constructor is risky. Operations on two objects will be reflecting on the same reference variable and cause unexpected behavior. This urges the need to implement copy constructor on your own when needed for creating two sets of resources. (**deep copy**)

```
39  String::String(const String& other)
40      : m_size(other.m_size)
41  {
42      m_text = new char[m_size+1];
43      strcpy(m_text, other.m_text);
44  }
```

Assignment operator

- When is it called?
  - Assignment operator is called when the existing object of a class being set equal to another object of the same class
    - A a1, a2;
    - a1 = a2;
- How it works?
  - By default: copy all the reference over (**shallow copy**)
  - Redefine: based on the code logic, typically makes a copy of dynamically allocated memory (**deep copy**)

Deep copy

1. Write the function signature
2. Clean up yourself (delete all the dynamically memory you have)
3. Allocate as much memory as the object to be copied
4. Copy all data over
5. return *this

```
51  String& String::operator=(const String& rhs)
52  {
53      if (this != &rhs)
54      {
55          String temp(rhs);
56          swap(temp);
57      }
58      return *this;
59  }
```

Initializer list

- A different way to assign values to class members.
  - If one member is an object of another class type, you have to initialize it with initializer list.
  - Data members in the initializer list is first constructed when calling the constructor.
- https://replit.com/@jiajerry/Order-of-Construction#main.cpp

Order of construction

- Construction Order:
  - member variable ---> member functions
  - Initializer prior to code in constructor
- https://replit.com/@jiajerry/Order-of-Construction#main.cpp

Worksheet problems

- http://web.cs.ucla.edu/classes/spring21/cs32/