C++ 프로그래밍

김 형 기

hk.kim@jbnu.ac.kr

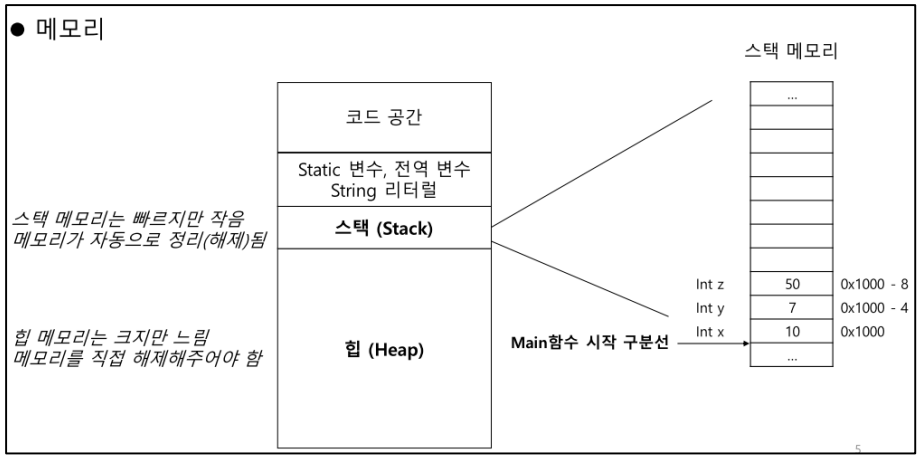
Summary of 2023-2 C++ Programming Class

- Introduction
  - C++ 탄생 배경, 출력 방법, 컴파일/링킹이 무엇인지, 전처리기(#), namespace
  - 변수
    - Bit, Byte, Hex, 변수와 변수의 주소란 무엇인지, VS에서 메모리 상태 관찰 방법, 변수의 사용과 정의
  - VS에서의 디버깅 방법(중단점, step over(F10), step in(F11))
- 배열/벡터, 명령문, 연산자, 제어문(조건문, 반복문)
- 함수
  - 독립적인 연산으로 분할, 모듈화하여 재사용성 확보, 프로토타입, 오버로딩
  - stack & heap, 함수를 호출하면 구분선이 생기고, 끝나면 지역변수가 없어짐
  - pass-by-value/address/reference의 차이점

문의사항: hk.kim@jbnu.ac.kr || diskhkme@gmail.com

Summary of 2023-2 C++ Programming Class

- 포인터
  - 주소를 저장하는 변수
  - 선언(*), 역참조(*), 주소값(&)
  - pass-by-address, 동적 할당(new, delete)은 heap 메모리
  - 허상 포인터, 메모리 누수에 유의
- 참조자
  - 변수의 별명
  - 선언(&)과 동시에 초기화, pass-by-reference
  - const 참조자로 복사 비용을 줄인다



문의사항: hk.kim@jbnu.ac.kr || diskhkme@gmail.com

Summary of 2023-2 C++ Programming Class

- 객체지향 프로그래밍
  - 절차적 프로그래밍과 다르게 데이터와 기능을 클래스로 캡슐화
- 클래스와 객체
  - 클래스는 사용자 정의 자료형, 데이터(멤버 변수)와 기능(멤버 함수)을 정의
  - ., ->로 접근, public/private/protected
  - 생성자, 소멸자
  - 생성자 오버로딩, 생성자 초기화 리스트, 생성자 위임
  - (얕은/깊은) 복사 생성자
  - this, const, static, friend
  - 멤버 변수 값을 변경하지 않는 멤버 함수는 함수 뒤에 const 붙여주어야 함

문의사항: hk.kim@jbnu.ac.kr || diskhkme@gmail.com

Summary of 2023-2 C++ Programming Class

- 상속
  - 기본 클래스를 기반으로 새로운 유도 클래스를 생성
  - 유도 클래스는 기본 클래스를 포함(멤버 변수/함수)
  - 상속하면 타입이 두개가 됨. 기본 클래스를 사용하는 곳에는 유도 클래스도 사용 가능하다
  - protected 멤버는 상속 되는 private 개념
  - 기본 클래스 생성자→유도 클래스 생성자 순으로 호출, 소멸자는 반대 순서
  - 유도 클래스 생성자에서 기본 클래스의 어떤 생성자를 호출할지 명시 가능
  - Base *ptr = new Derived(); 가능 하지만, 기본적으로 정적 바인딩

문의사항: hk.kim@jbnu.ac.kr || diskhkme@gmail.com

Summary of 2023-2 C++ Programming Class

- 다형성
  - C++에서 동적 바인딩 하려면? → 함수 오버라이딩
  - 1. 상속 2. 기본 클래스의 포인터 또는 참조자 3. 가상 함수 조건 필요
    - 기본 클래스의 가상(virtual) 멤버 함수를 유도 클래스에서 오버라이딩하면 동적 바인딩
    - 오버라이딩 할거면 가상 소멸자 만들어 줘야 함
  - override / final 지정자
  - 본문이 없는 가상 함수는 순수 가상 함수 (void func() = 0;)
  - 순수 가상함수가 포함된 클래스는 추상 클래스. 객체 생성 불가능
  - 순수 가상 함수만으로 이루어진 클래스는 인터페이스 클래스

문의사항: hk.kim@jbnu.ac.kr || diskhkme@gmail.com

Summary of 2023-2 C++ Programming Class

- 연산자 오버로딩
  - 멤버 함수 구현, 전역 함수(friend 사용) 구현 (p1 * 3, 3 * p1)
  - p1 + p2 → p1.operator+(p2) or operator+(p1,p2) 변환 방식을 이해해야 함
  - 스트림 삽입 및 추출, 대입 연산자, 첨자 연산자
  - 반환형, chaining을 위함 참조자 사용에 주의 (ostream& 반환)
  - 대입 연산자의 경우 얕은/깊은 복사 주의
- 템플릿
  - Generic programming을 위해서 매크로 또는 템플릿 사용 가능
  - template <typename T> 선언 후 T 사용하여 함수/클래스 템플릿 구현
  - T가 사용자 정의 타입일 경우 연산자 사용이 가능한지 체크 필요
  - 클래스 템플릿의 경우 반환형, 파라메터 등 적절한 위치에 T 사용 필요
  - 템플릿의 특수화, 템플릿의 인자<int N>

문의사항: hk.kim@jbnu.ac.kr || diskhkme@gmail.com

Summary of 2023-2 C++ Programming Class

- STL
  - 컨테이너, 알고리즘, 반복자
  - 반복자는 포인터와 유사. 역참조를 통해 값에 접근(begin(), end() → 반복자 반환)
  - 알고리즘 사용에 functor/function pointer/lambda expression 필요한 경우 있음
  - Sequence 컨테이너
    - array, vector, list
    - list는 첨자 연산자로 접근 불가
  - Associative 컨테이너
    - set, map
    - key-value pair

문의사항: hk.kim@jbnu.ac.kr || diskhkme@gmail.com

Summary of 2023-2 C++ Programming Class

- 더 하지 못한 내용…
  - 예외 처리
    - 예외상황에서의 동작 처리를 위한 Try, catch문, stack unwinding
  - IO Stream
    - C++의 입출력 스트림 조작 기능, filestream, stringstream
  - 스마트 포인터
    - 포인터 관리를 위한 부가 기능, unique_ptr, shared_ptr, weak_ptr

문의사항: hk.kim@jbnu.ac.kr || diskhkme@gmail.com

수고 많으셨습니다!