**CPP 2주차 (함수~)**

2022/1/3 ~ 2022/1/

**Do.**

1. **함수**

* 함수의 선언, 정의, 사용방법
* 매개변수의 디폴트 값 : 매개변수가 지정되지 않을 때 정해진 값을 대입하여 사용함.
* Call by Value와 Call by Reference
  + Call by Value : 값에 의한 호출로 값이 복제되어 적용되기 때문에 원본에 영향이 없다.
  + Call by Reference : 주소에 의한 호출로 원본의 메모리주소를 통해 기능이 수행되기 때문에 원본에 영향을 준다.-> 참조자(데이터타입&) 사용
  + 참조자는 일종의 변수의 별명 짓는 방법이다. 그러므로 같은 변수를 가리킨다.
  + Vector같은 배열형태의 변수도 똑같이 작용한다.
* 메모리의 힙 영역과 스택 영역
  + 힙 영역 : 사용자의 동적할당 ( 런 타임의 크기가 결정된다. )
  + 스택 영역 : 지역변수, 매개변수
* 함수를 이용한 계산기 만들기 – 과제

1. **배열**
   * 배열은 같은 데이터 타입의 집합이고 메모리상에서 연속적이다.
   * const 키워드 : 상수로 지정하는 함수로 코드상에서 변경할 수 없다.
   * 배열은 map처럼 사용이 가능하다. 숫자와 문자를 매칭해서 사용한다.
   * \*(arr + i) == arr[i]는 같은 기능을 수행한다. -> 주소의 값을 가리킴.
   * 배열은 메모리상에서 연속적이기 때문에 칸을 구분하지 않고 값을 입력해도 크기에 맞춰 알아서 대입된다. int arr2[2][5] = {1,2,3,4,5,6,7,8,9,10};
   * 이차원배열에서 포인터를 사용하기 위해서는 \*\*arr + i의 형태로 사용해야한다.
   * arr == &arr[0], \*arr == &arr[0][0], arr[1][2] == \*(\*(arr+1)+2)
2. **포인터**
   * 포인터 변수는 주소를 담는 변수이다.
   * 포인터 변수 앞에 붙는 \*연산자는 해당 변수가 가리키는 값을 가져오라는 의미이다.
   * 포인터 변수의 값으로는 참조하는 변수의 주소를 가리키므로 주소에 의한 호출과 같은 역할을 한다.
   * 결국 포인터 변수도 Call By Reference의 하나로 볼 수 있다.
3. **구조체**
   * 서로 다른 데이터타입의 묶음은 구조체.
   * C++에서의 구조체는 상속 가능하며 생성자 및 함수가 사용 가능하다.
   * 구조체는 매개변수로 사용 가능하다.
   * 사용자 정의 데이터 타입으로 불린다.
   * 클래스와 차이 : 접근제한자가 public으로 고정됨.
   * 객체를 반환하는 함수는 찾아낸 객체의 주소를 보내 객체의 포인터형으로 반환하면 Call by Reference의 형태로 반환되어 쓸모없는 값의 복제가 일어나지 않고 반환한 객체의 포인터를 통해 값을 수정할 수 있게 된다.