Friday, March 26, 2021

HW5: due Thursday 4/1

Today:

- brief HW5 overview ⇒ maps
- C++ references & dynamic allocation
- ex 9-1 review
- ex 9-2

slido.com
⇒ jhuip02 ⇐
zero

Recap Qs

1) 2)

Reference: alias for a variable
||
alternate name

Why? – reference parameter
– modify argument
– avoid copying
ALWAYS use reference param rather passing object by value

```
vector<double> large;
```

read lots of data

```
double sum = vecsum(large);
```

```
double vecsum(vector<double> vec) {
```

("by value")

copying

alias (no copying!)

```
double vecsum(const vector<double> & vec) {
```

```
void fill(vector<double> &vec,
          unsigned n, double v) {

    for ( int i=0; i<n; i++) {
        vec.push_back( v );
    }
}
```

alias

```
vector<double> data;
fill(data, 100, 42.0);
```

3) reference - convenience (no explicit dereference)
once created, cannot change what refers to

pointer - less convenient
can change what pointer variable points to

```
int a = 3;
int &b = a;   // b is alias for a
b = 4;
cout << a;    // prints "4"
int c = 9;
b = c;        // assign 9 to a
cout << a;    // prints "9"
```

4,5) allocate : new

dellocate : delete (single object) or variable

delete[] (array of objects)

Why? — constructor / destructors

Also: no sizeof

```
string s = "hello";

cout << s[1];        // print "e"

s[0] = 'J';

cout << s;           // prints "Jello"
```

text

Foo␣ bar␣ baz

in (an istream)

```
string s;
                return ref to in
while (in >> s) {
    // s is a token
}
```