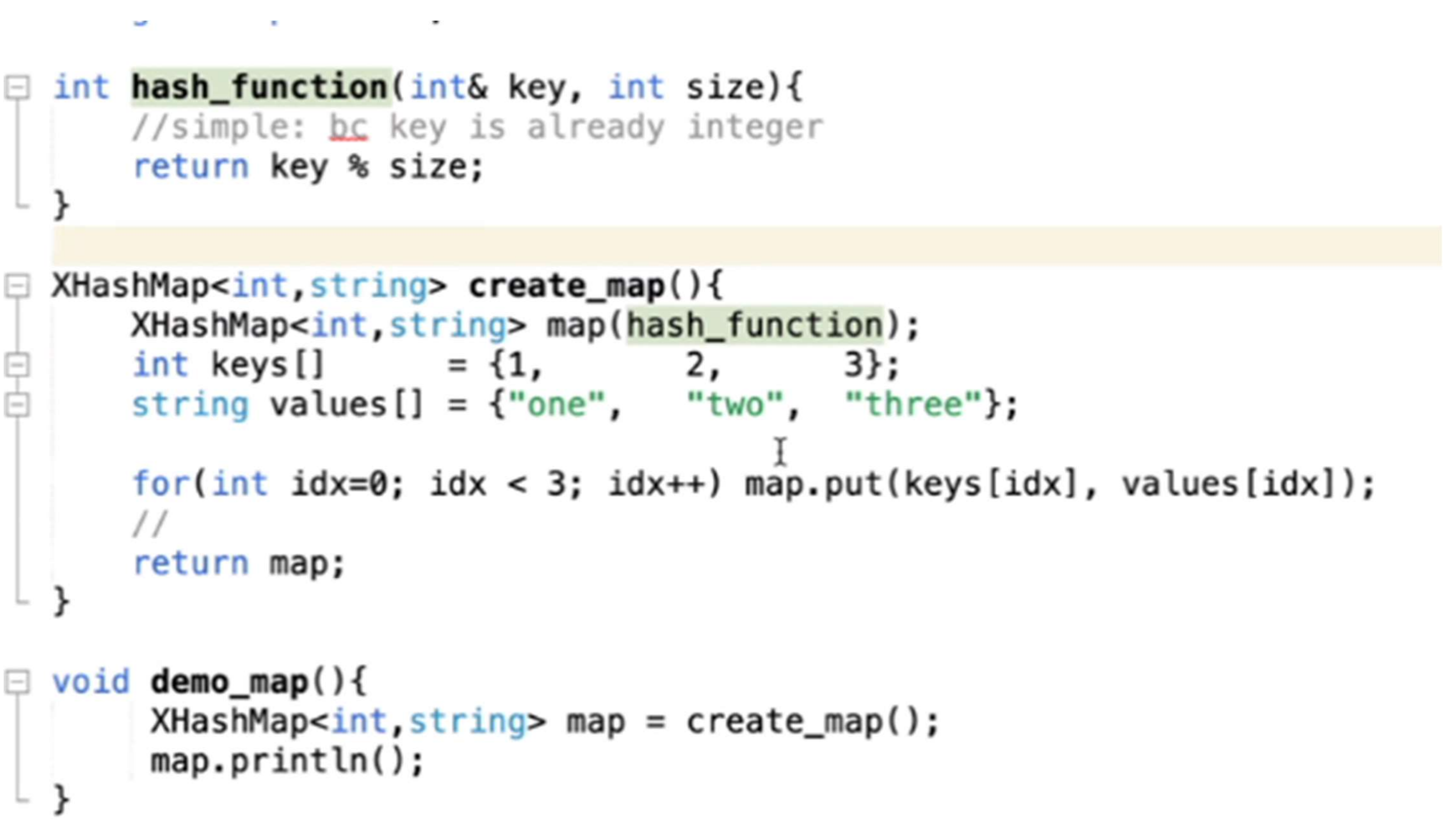
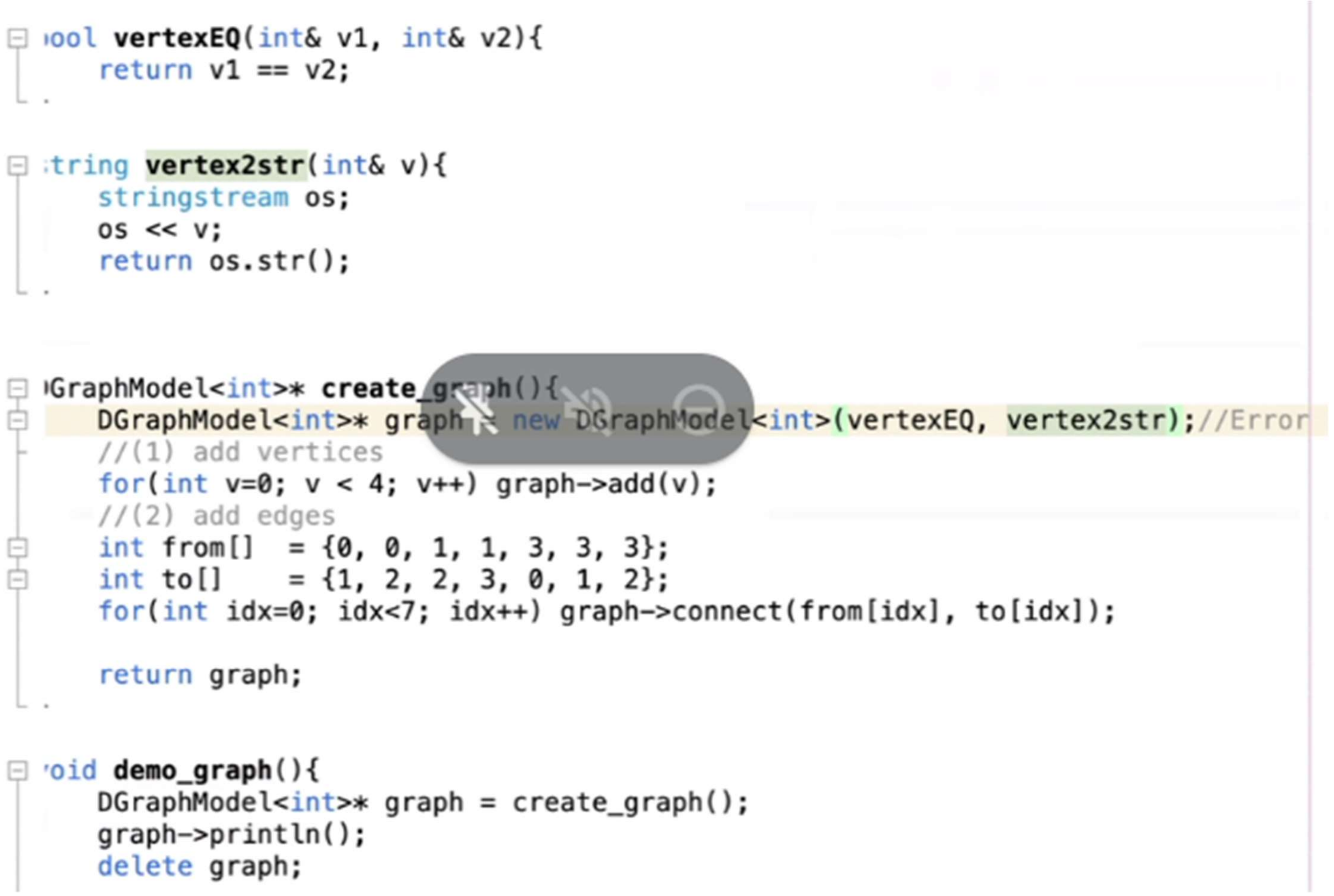


```
int hash_function(int& key, int size){
    //simple: bc key is already integer
    return key % size;
}

XHashMap<int,string> create_map(){
    XHashMap<int,string> map(hash_function);
    int keys[]      = {1,       2,       3};
    string values[] = {"one",   "two",   "three"};

    for(int idx=0; idx < 3; idx++) map.put(keys[idx], values[idx]);
    //
    return map;
}

void demo_map(){
     XHashMap<int,string> map = create_map();
     map.println();
}
```



```
ool vertexEQ(int& v1, int& v2){
    return v1 == v2;

:tring vertex2str(int& v){
    stringstream os;
    os << v;
    return os.str();

GraphModel<int>* create_graph(){
    DGraphModel<int>* graph = new DGraphModel<int>(vertexEQ, vertex2str);//Error
    //(1) add vertices
    for(int v=0; v < 4; v++) graph->add(v);
    //(2) add edges
    int from[]  = {0, 0, 1, 1, 3, 3, 3};
    int to[]    = {1, 2, 2, 3, 0, 1, 2};
    for(int idx=0; idx<7; idx++) graph->connect(from[idx], to[idx]);

    return graph;

'oid demo_graph(){
    DGraphModel<int>* graph = create_graph();
    graph->println();
    delete graph;
```