

ECE 220: Computer Systems & Programming

Lecture 23: C++ Exmples Thomas Moon

April 16 , 2024



Get the example codes

- https://github.com/tmoon-illinois/ece220_sp24/tree/main/lec23

1. Fix the bugs

vehicle.cpp

1. Fix the bugs

vehicle.cpp

- To access the private members, the class should provide public member functions.
- Naming convention
 - get... → return the private member (for reading)
 - set... → set the private member by parameters (for writing)

2. Use **STL vector** to replace array

vehicle_STL_vector.cpp

2. Use **STL vector** to replace array

vehicle_STL_vector.cpp

```
#include <vector> ← Need this to use STL vector
```

```
vector<Vehicle*> vlist;
```

```
vlist.push_back(v);
```

```
for(int i=0; i<vlist.size(); i++)
```

```
    vlist[i]->ShowData(); ← [ ] available in STL vector
```

3. Use **STL list** to replace vector

vehicle_STL_list.cpp

3. Use **STL list** to replace vector

vehicle_STL_list.cpp

```
#include <list> ← Need this to use STL vector
```

```
list<Vehicle*> vlist;
```

```
vlist.push_back(v);
```

```
for(int i=0; i<vlist.size(); i++)
```

```
—— vlist[i]->ShowData(); ← [ ] NOT available in STL list
```

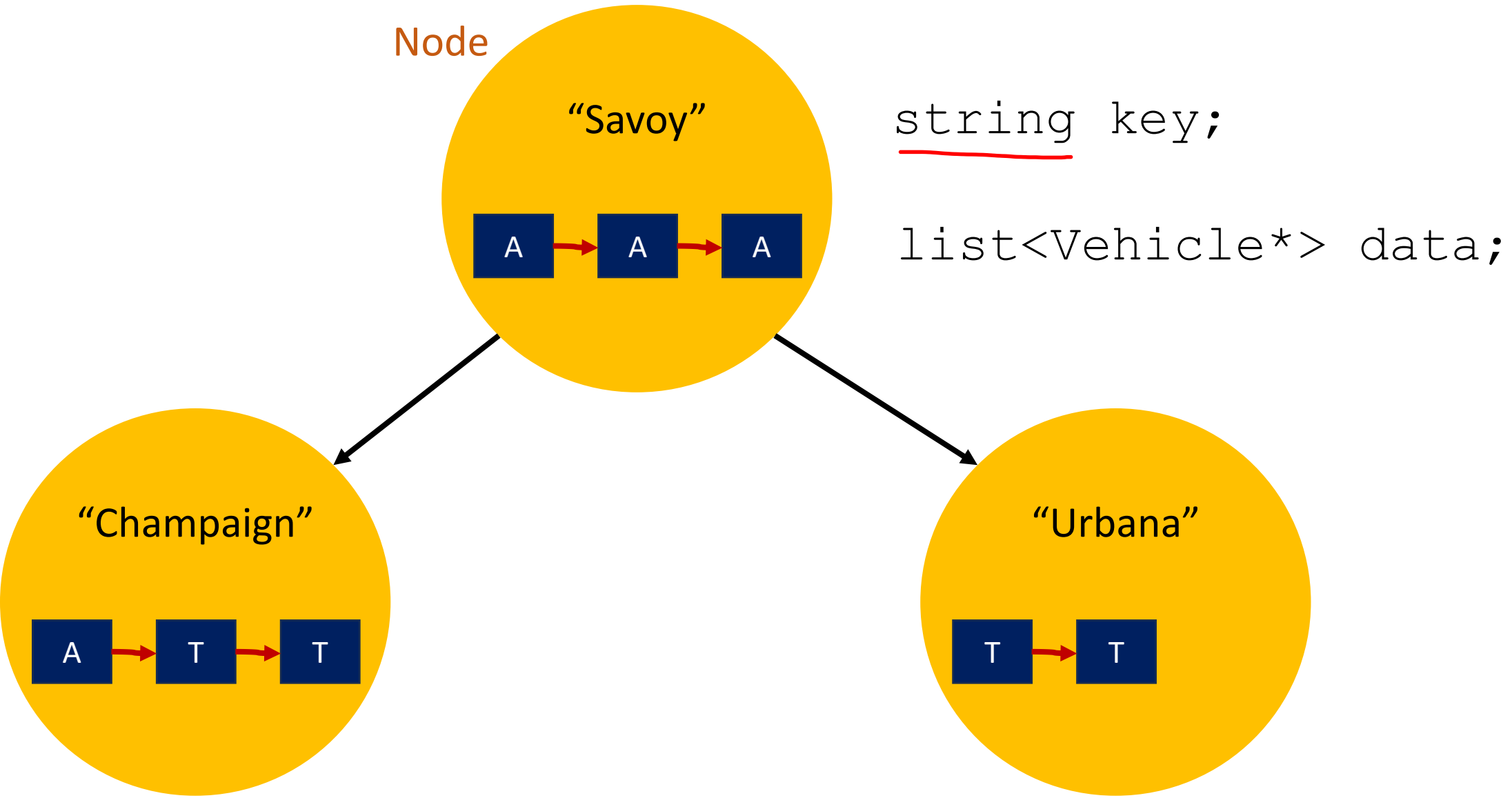
```
list<Vehicle*>::iterator it;
```

```
for(it=vlist.begin(); it!= vlist.end(); it++) {
```

```
    (*it)->ShowData();
```

```
}
```


4. Use binary search tree for City+Vehicle vehicle_STL_tree.cpp



string class

```
#include <string>
```

```
string str1("ABC"); ←  
string str2 = "123"; ←  
string str3 = str1 + str2; ←
```

```
cout<<"str1: "<<str1<<endl;  
cout<<"str2: "<<str2<<endl;  
cout<<"str3: "<<str3<<endl;
```

```
string str1, str2;  
str1 = "ABC"; ←  
str2 = "DEF"; ←
```

```
cout<<(str1==str2)<<endl;  
cout<<(str1<str2)<<endl;
```

```
str1: ABC  
str2: 123  
str3: ABC123
```

```
0  
1
```

*Warning 
≠ String in C (array of characters)

Write Node class member functions

```
class Node {
    private:
        string key;
        list<Vehicle*> data;
    public:
        Node *left;
        Node *right;

        Node(string key_, list<Vehicle*> data_);
        string getKey();
        void setKey(string key_);
        list<Vehicle*> getData();
        void setData(list<Vehicle*> data_);
};
Node::Node(string key_, list<Vehicle*> data_){
}
string Node::getKey(){
}
void Node::setKey(string key_){
}
list<Vehicle*> Node::getData(){
}
void Node::setData(list<Vehicle*> data_){
}
```

Write BinaryTree class member functions

```
class BinaryTree{
    private:
        Node *root;
    public:
        BinaryTree();
        ~BinaryTree(){};
        Node* getRoot(){return root;};
        void insert_node(Node *root, Node *data);
        void post_order(Node *root);
};
BinaryTree::BinaryTree(){
}
void BinaryTree::post_order(Node *node){
}
void BinaryTree::insert_node(Node *node, Node *data){
}
```