

```
void traversal(Visitor& v, Tree& t) {  
    ...  
    t.accept(v);  
    ...  
}
```

C++

Dynamic
Dispatch

```
void traversal(Visitor& v, Tree& t) {
```

```
...
```

```
t.accept(v)
```

```
...
```

```
}
```

```
class literal : public Tree {  
    virtual void accept(Visitor& v) {  
        v.visit(*this)
```

```
// static type
```

```
// this: literal*
```

```
}
```

```
};
```

Dynamic
Dispatch

```
void traversal(Visitor& v, Tree& t) {
```

```
...
```

```
t.accept(v)
```

```
...
```

```
}
```

```
class literal : public Tree {  
    virtual void accept(Visitor& v) {
```

```
        v.visit(*this);
```

```
        // static ...  
        // this: ...
```

```
    }  
};
```

```
class HeightVisitor : public Visitor {  
    virtual void visit(literal& N) {
```

```
        N.h = 1;
```

```
    }  
};
```

Dynamic
Dispatch