Reference

February 13, 2017

1 Compound Types

A compound type is a type that is defined in terms of another type. Simple declarations consist of a type followed by a list of variable names. More generally, a declaration is a base type followed by a list of declarators. Each declarator names a variable and gives the variable a type that is related to the base type.

2 References

Technically speaking, when we use the term reference, we mean "lvalue reference.

A reference defines an alternative name for an object. A reference type "refers to another type. We define a reference type by writing a declarator of the form &d, where d is the name being declared.

```
int ival = 1024;
int &refVal = ival; // refVal refers to (is another name for) ival
int &refVal2; // error: a reference must be initialized
```

Ordinarily, when we initialize a variable, the value of the initializer is copied into the object we are creating. When we define a reference, instead of copying the initializers value, we bind the reference to its initializer. Once initialized, a reference remains bound to its initial object. There is no way to rebind a reference to refer to a different object. Because there is no way to rebind a reference, references must be initialized.

A Reference Is an Alias

A reference is not an object. Instead, a reference is just another name for an already existing object. After a reference has been defined, all operations on that reference are actually operations on the object to which the reference is bound.

refVal = 2; // assigns 2 to the object to which refVal refers, i.e., to ival

int ii = refVal; // same as ii = ival

When we assign to a reference, we are assigning to the object to which the reference is bound. When we fetch the value of a reference, we are really fetching the value of the object to which the reference is bound. When we use a reference as an initializer, we are really using the object to which the reference is bound.

// ok: refVal3 is bound to the object to which refVal is bound, i.e., to ival

int &refVal3 = refVal;

Because references are not objects, we may not define a reference to a reference.

Reference Definitions

We can define multiple references in a single definition. Each identifier that is a reference must be preceded by the & symbol.

With two exceptions, the type of a reference and the object to which the reference refers must match exactly. A reference may be bound only to an object, not to a literal or to

the result of a more general expression

int &refVal4 = 10; // error: initializer must be an object

double dval = 3.14;

int &refVal5 = dval; // error: initializer must be an int object

2.1 rvalue reference

These references are primarily intended for use inside classes.