**Feb 15, 2021**

1. **Inheritance (continued)**

**class A : protected B**

**{**

**};**

**Difference between private and protected data members in a base class.**

**Can we inherit everything from the base class?**

**No. There are three things that can’t be inherited:**

1. **Constructors**
2. **“=” operator**
3. **friend**

**If class A is a friend of B, it does not mean a subclass of A is a friend of B.**

1. **Operator Overloading**

**#define DIM 10**

**class Row**

**{**

**private:**

**Int x[DIM]; // valid index range: [0, 9]**

**public:**

**Row() { for(int i=0; i<DIM; i++)**

**x[i]=0; }**

**Row(int y[])**

**{**

**for(int i=0; i<DIM; i++)**

**y[i]=x[i]; // Wrong:R.H.S 🡪 L.H.S.**

**x[i]=y[i]; // correct**

**}**

**Row(Row &w) // copy constructor**

**{**

**for(int i=0; i<DIM; i++)**

**x[i] = w.x[i];**

**}**

**Row& operator = (Row &w)**

**{**

**for(int i=0; i<DIM; i++)**

**x[i] = w.x[i];**

**return \*this;**

**}**

**Row& operator += (Row &w)**

**{**

**for(int i=0; i<DIM; i++)**

**x[i] += w.x[i];**

**return \*this;**

**}**

**bool operator == (Row &left, Row &right)**

**{**

**for(int i=0; i<DIM; i++)**

**if(left.x[i] != right.x[i])**

**return false;**

**return true;**

**}**

**// prefix ++ operator: do ++ first, and then =**

**Row& operator ++( )**

**{**

**for(int i=0; i<DIM; i++)**

**x[i]++;**

**return \*this;**

**}**

**// postfix ++ operator: do “=” first, and then ++**

**Row operator ++(int )**

**{**

**Int y[DIM];**

**for(int i=0; i<DIM; i++)**

**y[i]=x[i];**

**Row t(y);**

**for(int i=0; i<DIM; i++)**

**x[i]++;**

**return t;**

**}**

**// ostream is a friend of Row such that it can**

**// output the private data member of Row**

**friend ostream operator << (ostream &os,**

**const Row &w)**

**{**

**for(int i=0; i<DIM; i++)**

**os << w.x[i] << “ “ ;**

**}**

**};**

1. **List**

**It is an important data structure in computer science.**