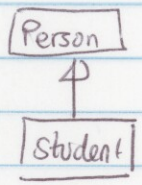
ECE 462
10/8/2010




```
ostream& operator << (ostream& os, const Person& p) {
    os << p.getName() << endl;
    return os;
}
```

```
ostream& operator << (ostream& os, const Student& s) {
    const Person* p = &s;
    os << (*p);
    os << s.getSchool();
    return os;
}
```

address

pointer, address

dereference

take p's value as address and read from that address

```
int a = 6
int *b = &a;
int c = *b;
// c = 6
```

```
class Point3D;    //forward declaration
class Point2D {
  private:
    int x;
    int y;
  public:
    Point2D(int a, int b) {
      x = a;
      y = b;
    }
    operator Point3D();
};
```

```
class Point3D {
  private:
    int x;
    int y;
    int z;
  public:
    Point3D(int a, int b, int c) {
      x = a;
      y = b;
      z = c;
    }
    Operator int() const {
      return (x+y+z);
    }
    friend operator<<
    ostream& operator << (ostream& os,
      const Point3D& p) {
      os << p.x << " " << p.y << " " << p.z << endl;
      return os; }
```

no type

// convert point2D → point3D

```
Point2D::operator Point3D() {
  return (x, y, 0);
}
```

Point3D

return type