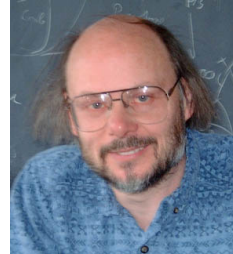


C++

CSCI 334
Stephen Freund



Stroustrup Quotes

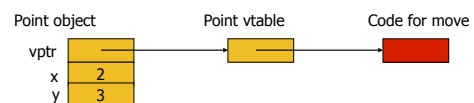
- "C makes it easy to shoot yourself in the foot; C++ makes it harder, but when you do, it blows away your whole leg."
- "I have always wished that my **computer** would be as easy to use as my **telephone**. My wish has come true. I no longer know how to use my **telephone**."

C++ Object System

- Classes & Objects
 - heap allocated or stack allocated
- Dynamic Lookup
 - "virtual" or "non-virtual"
- Inheritance
 - single or multiple. public or private.
- Subtyping
 - A <: B if B is public superclass of A
- Encapsulation
 - private, protected, public

```
class Point {  
private:  
    int x,y;  
public:  
    Point(int xv, int yv);  
    int getX();  
    int getY();  
    virtual void move(int dx, int dy);  
protected:  
    void setLocation(int xv, int yv);  
};  
  
Point::Point(int xv, int yv) { x = xv; y = yv; }  
int Point::getX() { return x; }  
int Point::getY() { return y; }  
void Point::setLocation(int xv, int yv) { x = xv; y = yv; }  
void Point::move(int dx, int dy) {setLocation(x+dx,y+dy);}
```

C++ Run-Time Representation



```

class ColorPoint: public Point {
private:
    int c;
public:
    ColorPoint(int xv, int yv, int cv);
    int getColor(void);
    virtual void move(int dx, int dy);
    virtual void darken(int tint);
};

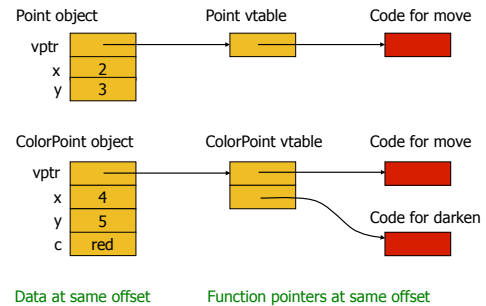
ColorPoint::ColorPoint(int xv, int yv, int cv):
    Point(xv, yv)
    { c = cv; }

int ColorPoint::getColor(void) { return c; }

void ColorPoint::move(int dx, int dy)
    { Point::move(dx, dy); darken(1); }

```

C++ Run-Time Representation



Run-time Representation

