CSE 333    13AU
SECTION 5

References, Classes and const.
This or that?

• Consider the following code:

**Pointers:**
```c
int i;
int *pi = &i;
```

**References:**
```c
int i;
int &ri = i;
```

In both cases,

```c
*pi = 4;
ri = 4;
```

The difference lies in how they are used in expressions:
C++ const declaration

- As a declaration specifier, const is a type specifier that makes objects unmodifiable.
  ```cpp
  int const m = 255;
  ```

- Reference to constant integer:
  ```cpp
  int n = 100;
  int const &ri = n;  // ri becomes read only
  ```
When to use?

• Function parameter types and return types and functions that declare overloaded operators.

• **Pointers**: may point to many different objects during its lifetime. Pointer arithmetic (++ or --) enables moving from one address to another. (Arrays, for e.g.)

• **References**: can refer to only one object during its lifetime.

• **Style Guide Tip**:  
  • use const reference parameters to pass input  
  • use pointers to pass output parameters  
  • input parameters first, then output parameters last
Example – Pass by Reference (Recap)

```c
void swap(int &x, int &y) {
    int tmp = x;
    x = y;
    y = tmp;
}

int main(int argc, char **argv) {
    int a = 5, b = 10;
    swap(a, b);
    cout << "a: " << a << "; b: " << b << endl;
    return EXIT_SUCCESS;
}
```
C++ Classes

//use const to protect class members (make get_x() read-only)
class Point {
public:
Point(const int x, const int y); // constructor
int get_x() { return x_; } // inline member function
int get_y() { return y_; } // inline member function
double Distance(const Point &p) const; // member function
void SetLocation(const int x, const int y); // member function

private:
int x_; // data member
int y_; // data member
}; // class Point
C++ Classes

class Point {

public:
Point(const int x, const int y); // constructor
int const get_x() { return x_; } // inline member function
int const get_y() { return y_; } // inline member function
double Distance(const Point &p) const; // member function
void SetLocation(const int x, const int y); // member function

private:
int x_; // data member
int y_; // data member
}; // class Point
class Point {
public:
    Point(const int x, const int y); // constructor
    int const get_x() { return x_; } // inline member function
    int const get_y() { return y_; } // inline member function
    double Distance(const Point &p) const; // member function
    void SetLocation(const int x, const int y); // member function

private:
    int x_; // data member
    int y_; // data member
}; // class Point
class Point {
  public:
  Point(const int x, const int y); // constructor
  int get_x() const { return x_; } // inline member function
  int get_y() const { return y_; } // inline member function
  double Distance(const Point &p) const; // member function
  void SetLocation(const int x, const int y); // member function

  private:
  int x_; // data member
  int y_; // data member
}; // class Point

const should be put:
  after function header get_x() and before{}