Person.cpp

Copy Constructor and Assignment Operator

Examples with C-Strings: Array and Pointer

Delroy A. Brinkerhoff

Person CLASS
ARRAY VERSION

```
class Person
{
    private:
        char    name[100] = "";
        int     weight = 0;
        double  height = 0;

    public:
        Person() {}
        Person(char* n, int w, double h) : weight(w), height(h) { strcpy(name, n); }
        Person(const Person& p);
        Person& operator=(Person& p);
};
```

Person CLASS
POINTER VERSION

```
class Person
{
    private:
        char*   name = nullptr;
        int     weight = 0;
        double  height = 0;
    public:
        Person() {}
        Person(char* n, int w, double h)
            : name(strcpy(new char[strlen(n)+1], n)), weight(w), height(h) {}
        Person(const Person& p);
        Person& operator=(Person& p);
};
```

INITIALIZING A CHARACTER POINTER

- `name(strcpy(new char[strlen(n) + 1], n))`
  - `strlen(n) + 1`
  - `new char[strlen(n) + 1]`
  - `strcpy(new char[strlen(n) + 1], n)`
  - `name(strcpy(new char[strlen(n) + 1]))`

THE COPY CONSTRUCTOR

ARRAY VERSION

```
Person::Person(const Person& p)
{
    memcpy(this, &p, sizeof(Person));
    /*strcpy(name, p.name);
    weight = p.weight;
    height = p.height;*/
}
```

POINTER VERSION

```
Person::Person(const Person& p)
{
    memcpy(this, &p, sizeof(Person));
    name = new char[strlen(p.name)+1];
    strcpy(name, p.name);
    //weight = p.weight;
    //height = p.height;
}
```

THE ASSIGNMENT OPERATOR

ARRAY VERSION

```
Person& Person::operator=(Person& p)
{
    if (this == &p)
        return *this;
    memcpy(this, &p, sizeof(Person));
    /*strcpy(name, p.name);
    weight = p.weight;
    height = p.height;*/
    return *this;
}
```

POINTER VERSION

```
Person& Person::operator=(Person& p)
{
    if (this == &p)
        return *this;
    memcpy(this, &p, sizeof(Person));
    name = new char[strlen(p.name)+1];
    strcpy(name, p.name);
    //weight = p.weight;
    //height = p.height;
    return *this;
}
```

USING THE COPY OPERATIONS

COPY CONSTRUCTOR

- Person p1(…);

- Person p2(p1);
- Person p3 = p1;

ASSIGNMENT OPERATOR

- Person p1(…);

- Person p2;
- p2 = p1;