Programming Portfolio

Technical Skills and Code Samples

Technical Summary

I use programming to support educational design, build organized tools, and structure content. My current skills include C plus plus, Java, HTML, CSS, and GitHub for version control and deployment.

Languages and Tools

- C++: pointers, classes, linked lists, memory management basics
- Java: object oriented programming, constructors, validation, simple data applications
- Web Design: HTML, CSS, responsive layouts, semantic structure, simple components
- GitHub: version control, commits, branches, GitHub Pages static site hosting

Sample C++ Code:

```
// Person.cpp
//

#include "Person.h"

void Person::print()
{
	printf("%s %s %s %s %s %s\n", firstName->c_str(), lastName->c_str(), street->c_str(), city->c_str(), state->c_str(), zip->c_str());
}

Person::Person(std::string* firstName, std::string* lastName, std::string* street, std::string* city, std::string* state, std::string* zip)
{
	this->firstName = NULL;
	this->lastName = NULL;
	this->street = NULL;
	this->city = NULL;
```

```
	this->state = NULL;
	this->zip = NULL;

	if (firstName)
		this->firstName = new std::string(*firstName);
	if (lastName)
		this->lastName = new std::string(*lastName);
	if (street)
		this->street = new std::string(*street);
	if (city)
		this->city = new std::string(*city);
	if (state)
		this->state = new std::string(*state);
	if (zip)
		this->zip = new std::string(*zip);


}
Person::~Person()
{
	if (firstName != nullptr)
		delete firstName;
	if (lastName != nullptr)
		delete lastName;
	if (street != nullptr)
		delete street;
	if (city != nullptr)
		delete city;
	if (state != nullptr)
		delete state;
	if (zip != nullptr)
		delete zip;
}
```

Sample Java Code:

```
/**
 * The Person class represents a person
 * with a name, address, and phone number.
 */

public class Person implements Comparable<Person>{
    private String name;
    private String address;
    private long phoneNumber;

    // Full-argument constructor
    public Person(String name, String address, long phoneNumber) {
        this.name = name;
        this.address = address;
        this.phoneNumber = phoneNumber;
    }

    // Copy constructor
    public Person(Person other) {
        this.name = other.name;
        this.address = other.address;
        this.phoneNumber = other.phoneNumber;
    }

    // Setters
    public void setName(String name) {
        this.name = name;
    }

    public void setAddress(String address) {
        this.address = address;
    }
```

```
    public void setPhoneNumber(long phoneNumber) {
        this.phoneNumber = phoneNumber;
    }

    // Getters
    public String getName() {
        return this.name;
    }

    public String getAddress() {
        return this.address;
    }

    public long getPhoneNumber() {
        return this.phoneNumber;
    }

    @Override
    public int compareTo(Person other) {
        return this.name.compareToIgnoreCase(other.name);
    }

    @Override
    public String toString() {
        return "Name: " + name + ", Address: " + address + ", Phone: " +
phoneNumber;
    }
}
```

For more information and live examples, visit my portfolio website.