CS2810 OOAIA

Functors

Example Code Snippets

1. **Basic Functor**

```cpp
#include <iostream>
using namespace std;

// Simple functor to square a number
class Square {
public:
    int operator()(int x) {
        return x * x;
    }
};

int main() {
    Square sq;
    cout << sq(5) << endl;  // Using functor like a
function
    return 0;
}
```

2. **Functor with State**

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
```

```cpp
using namespace std;

// Functor that keeps track of how many numbers are above
a threshold
class CountAbove {
private:
    int threshold;
    int count;
public:
    CountAbove(int t) : threshold(t), count(0) {}
    void operator()(int x) {
        if (x > threshold) count++;
    }
    int getCount() { return count; }
};

int main() {
    vector<int> nums = {1, 5, 3, 8, 2, 9};
    CountAbove counter(4);
    for_each(nums.begin(), nums.end(), counter);
    cout << "Numbers above 4: " << counter.getCount() <<
endl;
    return 0;
}
```

3. **Functor with Algorithm**

```cpp
#include <iostream>
#include <algorithm>
using namespace std;

// Functor to multiply each element by a factor
class Multiplier {
private:
    int factor;
public:
```

```
    Multiplier(int f) : factor(f) {}
    int operator()(int x) {
        return x * factor;
    }
};

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    transform(arr, arr+5, arr, Multiplier(3));
    for(int i = 0; i < 5; i++)
        cout << arr[i] << " ";
    return 0;
}
```

4. **String Processing Functor**

```
#include <iostream>
#include <string>
using namespace std;

// Functor to count specific characters in a string
class CharCounter {
private:
    char targetChar;
public:
    CharCounter(char c) : targetChar(c) {}
    int operator()(const string& str) {
        int count = 0;
        for(char c : str)
            if(c == targetChar) count++;
        return count;
    }
};

int main() {
    CharCounter countA('a');
```

```
    cout << countA("banana") << endl;
    return 0;
}
```

5. **Sorting with Functor**

```
#include <iostream>
#include <algorithm>
#include <vector>
using namespace std;

// Functor for custom sorting
class CustomSort {
public:
    bool operator()(int a, int b) {
        return (a % 10) < (b % 10);  // Sort by last digit
    }
};

int main() {
    vector<int> nums = {23, 45, 12, 89, 32};
    sort(nums.begin(), nums.end(), CustomSort());
    for(int n : nums)
        cout << n << " ";
    return 0;
}
```

6. **Filtering with Functor**

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

// Functor to filter numbers in a range
class RangeFilter {
```

```
private:
    int low, high;
public:
    RangeFilter(int l, int h) : low(l), high(h) {}
    bool operator()(int x) {
        return x >= low && x <= high;
    }
};

int main() {
    vector<int> nums = {5, 10, 15, 20, 25};
    int count = count_if(nums.begin(), nums.end(),
RangeFilter(10, 20));
    cout << "Numbers between 10 and 20: " << count <<
endl;
    return 0;
}
```

Practice Problems

1. Create a functor that checks whether a given number is a palindrome (i.e., reads the same forward and backward). Use it with std::count_ifto count how many palindrome numbers exist in a given array.

2. Implement a functor that computes the running average of the numbers it processes. Each time it is called with a new number, it should update the average and return the current computed value.

3. Design a functor that processes strings and counts words that satisfy certain criteria, such as words that start with a vowel or words of a specific length. Use it with standard algorithms like std::count_if or std::for_each to analyze a collection of strings.

4. Develop a functor that acts as a simple text parser. It should process strings and identify specific patterns such as dates, email addresses, or phone numbers. Use it with standard algorithms to extract meaningful information from a collection of strings (You can use Regex).