

Khoi Duong

Prof. Yang

CS360

6/28/2022

MIDTERM

1.

Source code:

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

class Money{
private:
    long double mny;
public:
    Money();
    Money(char const s[]);
    void mstold(char const s[]);
    void ldtoms();
    void madd(Money m1, Money m2);
    void getmoney();
    void putmoney();
};

// Constructor
Money::Money() {
    mny = 0;
}

Money::Money(char const s[]){
    mstold(s);
}
```

```

// Use the function strtold() to convert a money string entered as input into a long
double
void Money::strtold(char const s[]){
    mny = strtold(s);
}

// Use the function ldtoms() to convert a long double into a money string for
display
string ldtoms (long double x){
    string s = to_string(x);
    return s;
}

// Use the function getmoney() to get a money string from the user
void Money::getmoney(){
    char s[100];
    cout << "Enter a money string: ";
    cin >> s;
    mny = atof(s);
}

// Use the function putmoney() to display a money string to the user
void Money::putmoney(){
    cout << mny << endl;
}

// Use the function madd() to add two money strings
void Money::madd(Money m1, Money m2){
    mny = m1.mny + m2.mny;
}

int main(){
    Money m1, m2, m3;
    m1.getmoney();
    m2.getmoney();
    m3.madd(m1, m2);
    cout << "The sum of m1 and m2 is: "; m3.putmoney();

    Money m4("123.45");
    cout << "The money string m4 is: "; m4.putmoney();

    return 0;
}

```

Run program & result:

```

PS D:\VS CODE\C C++\CS360\Midterm> cd "d:\VS CODE\C C++\CS360\Midterm\"; if ($?) { g++ 1.cpp -o 1 } ; if ($?) { .\1 }
Enter a money string: 15.64
Enter a money string: 18.94
The sum of m1 and m2 is: 34.58
The money string m4 is: 123.45
PS D:\VS CODE\C C++\CS360\Midterm>

```

Github link: https://github.com/MynameisKoi/CS360/blob/main/midterm_1.cpp

2.

Source code:

```

#include <iostream>
#include <bits/stdc++.h>
using namespace std;

// define a function template interpolSearch() that looks up a given element in a
sorted numeric array. This array elements are of the same type as the template
parameter T
// the function template has three parameters - the value searched for of type T,
a pointer to the first array element, and the number of array elements

template <typename T>
int interpolSearch(T arr[], int n, T key){
    // sort the array elements by number
    sort(arr, arr + n);
    cout << "Array after sorting: " << endl;
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";

    // find the index of the element we are looking for
    int low = 0;
    int high = n - 1;
    while (low <= high){
        int mid = low + (high - low) / 2;
        if (arr[mid] == key)
            return mid;
        else if (arr[mid] < key)
            low = mid + 1;
    }
}

```

```

        else
            high = mid - 1;
    }
    return -1;
}

int main(){
    // create a random array of integers
    int size = 20;
    int arr[size];
    for (int i = 0; i < size; i++)
        arr[i] = rand() % 10;
    // print the array of integers
    for (int i = 0; i < size; i++)
        cout << arr[i] << " ";
    cout << endl;

    // verify interpolSearch()
    int n = sizeof(arr) / sizeof(arr[0]);
    int key = 5;
    int index = interpolSearch(arr, n, key);
    if (index == -1)
        cout << "Element not found" << endl;
    else
        cout << "Element found at index " << index << endl;
    return 0;
}

```

Run program & result:

```

> cd "d:\VS CODE\C C++\CS360\Midterm\"; if ($?) { g++ 2.cpp -o 2 } ; if ($?) { .\2 }
1 7 4 0 9 4 8 8 2 4 5 5 1 7 1 1 5 2 7 6
Array after sorting:
0 1 1 1 1 2 2 4 4 4 5 5 5 6 7 7 7 8 8 9 Element found at index 11
PS D:\VS CODE\C C++\CS360\Midterm>

```

Github link: https://github.com/MynameisKoi/CS360/blob/main/midterm_2.cpp

3.

Source code:

```

#include <iostream>
#include <math.h>
#include <string>
using namespace std;

class Poly{
private:
    int coeff;
    int exp;
    int array[10];
    // constructor takes coefficient and exponent as arguments
    // use an object to store exponents and coefficients
    // Poly c = Poly(3,4) + Poly(2,2) + Poly(7,1) + Poly(-5,0)
public:
    Poly(){
        for (int i = 0; i < 10; i++) {
            array[i] = 0;
        }
    }
    Poly(int c, int e){
        for (int i = 0; i < 10; i++){
            if (i == e) {
                array[i] = c;
            }
            else {
                array[i] = 0;
            }
        }
    }

    // overload += operator
    Poly operator+=(Poly rhs){
        for (int i = 0; i < 10; i++){
            array[i] += array[i] + rhs.array[i];
        }
        return *this;
    }

    // overload -= operator
    Poly operator-=(Poly &rhs){
        for (int i = 0; i < 10; i++){
            array[i] += array[i] - rhs.array[i];
        }
        return *this;
    }
}

```

```

// add two polynomials
Poly operator+(Poly p){
    Poly q;
    for (int i = 0; i < 10; i++){
        q.array[i] = array[i] + p.array[i];
    }
    return q;
}

// subtract two polynomials
Poly operator-(Poly p){
    Poly q;
    for (int i = 0; i < 10; i++){
        q.array[i] = array[i] - p.array[i];
    }
    return q;
}

// multiply polynomials by a constant
Poly operator*(int c){
    Poly q;
    for (int i = 0; i < 100; i++){
        q.array[i] = array[i] * c;
    }
    return q;
}

// evaluate polynomials at a specific value for x
int eval(int x){
    int sum = 0;
    for (int i = 0; i < 10; i++){
        sum += array[i] * pow(x, i);
    }
    return sum;
}

// display the polynomial
void display(){
    for (int i = 0; i < 10; i++){
        if (array[i] != 0){
            if (i == 0){
                cout << array[i];
            }
            else if (i == 1){

```

```

        cout << array[i] << "x";
    }
    else{
        cout << array[i] << "x^" << i;
    }
    cout << " + ";
}
}
cout << "0" << endl;
}
};

int main(){
    Poly p0(3,2);
    cout << "p0 = "; p0.display();
    Poly p1, p2;
    p1 += Poly(3,4) + Poly(2,2) + Poly(7,1) + Poly(-5,0);
    cout << "p1 = "; p1.display();
    p2 += Poly(1,1) + Poly(2,0) + Poly(3,3) + Poly(4,4);
    cout << "p2 = "; p2.display();
    Poly p3 = p1 + p2;
    cout << "Addition: "; p3.display();
    Poly p4 = p1 - p2;
    cout << "Subtraction: "; p4.display();
    Poly p5 = p1 * 2;
    cout << "Multiplication: "; p5.display();
    int x = 4;
    cout << "The value of p1 at x = " << x << " is: " << p1.eval(x) << endl;
    cout << "The value of p2 at x = " << x << " is: " << p2.eval(x) << endl;
    x = 7;
    cout << "The value of p3 at x = " << x << " is: " << p3.eval(x) << endl;
    x = 5;
    cout << "The value of p4 at x = " << x << " is: " << p4.eval(x) << endl;

    return 0;
}

```

Run program & result:

```
p0 = 3x^2 + 0
p1 = -5 + 7x + 2x^2 + 3x^4 + 0
p2 = 2 + 1x + 3x^3 + 4x^4 + 0
Addition: -3 + 8x + 2x^2 + 3x^3 + 7x^4 + 0
Subtraction: -7 + 6x + 2x^2 + -3x^3 + -1x^4 + 0
Multiplication: -10 + 14x + 4x^2 + 6x^4 + 0
The value of p1 at x = 4 is: 1656
The value of p2 at x = 4 is: -10
The value of p3 at x = 7 is: 14690
The value of p4 at x = 5 is: 150
PS D:\VS CODE\C C++\CS360\Midterm> 
```

Github link: https://github.com/MynameisKoi/CS360/blob/main/midterm_3.cpp