**Kerry Ojakian's CSI 32 Class**
**Due Date:**
Written: Wednesday March 13 in class; Coding: Thursday March 14 in Dropbox by 5pm

# HW #3

**General Instructions:** Write problems 1 to 3 on this paper and hand in at Wednesday March 13 class. Do not use a computer for problems 1 to 3; work them out by hand and show your work.
For problems 4 to 7 put them into ONE C++ file which you will put in a new folder HW03 in your Dropbox, in order: problem 4, then 5, then 6, then 7.
Follow the above instructions exactly!

# The Assignment

1. Suppose we have defined the function *G* as follows:

```
int G(int y, string st, vector<int> U) {
    y = 99;
    st[1] = ’r’;
    U[1] = 99;
    cout << "Inside: " << st << endl;
    return y;
```

What does the following print. Clearly indicate your answer (i.e. the print out) separate from your work.

```
    int y, z;
    y = 10;
    string s = "abcde";
    vector<int> W = {4,5,6,7};
    z = G(y, s, W);
    cout << y << endl;
    cout << s << endl;
    cout << W[1] << endl;
    cout << z << endl;
```

2. What does the following code print. Clearly indicate your answer (i.e. the print out) separate from your work.

```
    int x1 = 5;
    vector<char> V = {’x’, ’a’, ’d’};
    for (char ch : V) {
        cout << ch << endl;
        x1++;
    }
    cout << "Final value: " << x1 << endl;
```

3. What does the following code print. Clearly indicate your answer (i.e. the print out) separate from your work.

```
class Simple {
public:
    Simple(int y) {  x = 2*y;  }

    int get() { return x; }

    void change(int y) { x = y; }

    int x;
};

bool H(int x, Simple S) {
    x++;
    S.change(x);
    x = x + 10;
    cout << "Inside: " << S.get() << endl;
    return (x > 11);
}


int main() {
    int x = 9;
    Simple S(x+2);
    z = H(x, S);
    cout << "x:" << x << endl;
    cout << "S: " << S.get() << endl;
    cout << "z: " << z << endl;
}
```

4. Write a function `oddSum` which takes a single integer n as input (you can assume that $n$ is at least 1 and is odd). The function returns (not prints!) the sum of the odd integers from 1 to $n$. Do this in the direct way using loops. Then, if you can ... use math to do it in a very simple way without loops.

   Example: `oddSum`(5) should return 9, because $1 + 3 + 5 = 9$.

5. Write a function `smallFact` that takes a positive integer $n$ as input. You can assume that $n$ is a positive integer which is NOT prime. The function returns (not prints!) the smallest factor of $n$ that is larger than 1.

   Example: `smallFact`(12) should return 2 since the relevant factors of 12 are: 2, 3, 4, and 6 with 2 being the smallest factor.

6. Represent a point on the plane as a vector of floats of length 2. Write function named `slope` which takes two points as input (i.e. two float vectors as input). The function returns nothing, but prints as follows. It should print "Slope: " followed by the slope between the two points. You must account for the possibility that the line is vertical and so has an undefined slope, in that case print "Slope: Undefined".

   Example: `slope(` {2,3}, {3,5} ) should result in the printing of "Slope: 2"

7. Write a class called Point to represent a point on the plane (i.e. it should store an $x$ and a $y$ coordinate). Its methods are: 1) getX: returns the $x$ coordinate, 2) getY: returns the $y$ coordinate, 3) getQuad: Returns the quadrant or axis the point is on (i.e. either "I" or "II" or "III" or "IIII" or "X-Axis" or "Y-Axis" or "Both Axes").

   Then write a function that takes two Point objects as input and returns the slope between the 2 points (i.e. modify the last question).