Name: □ JHEID: □ Score: □

Homework 6

CAUTION

- You are expected to work individually.
- **Due: Friday November 13th at 11pm EST (Baltimore time).**
- *This assignment is worth 20 points.*

SUBMISSION REQUIREMENT

Answer each problem in this **pdf**, in the area to the side of the problem, or immediately after it. You may either type your solutions, or hand-write and scan them in, but they need to be legible, and part of this document. If you need to add additional sheets, please make a note near the problem itself that the grader should "see attached". Submit the **pdf** document via GradeScope once you have added your answers.

Learning Objectives

OBJECTIVES

- c++ file i/o
- standard template library
- c++ references
- classes and objects

INFO

Several problems make use of "code fragments," which can be thought of as pieces of code extracted from complete programs. While a code fragment will not generally compile by itself, we will assume that it exists in a sensible framework (i.e. is inside a properly formed `main()`, all appropriate headers and libraries have been included, etc.). We will also assume that there is no other code in the program that would impact the behavior of the fragment; each fragment is designed to be understood in isolation.

Part I: Code Tracing. [1 point each problem]

Trace through each code fragment and write down the exact output that will be printed if the fragment is run. If the value of some output generated cannot be determined, write **”unknown”** in place of that value.

1. For this question, assume the user inputs `cs220` and then `Ctrl-d` several times (to signal end-of-input) as the program runs:

```
int main() {
    char c;
    std::string s = "Welcome to ";
    while (std::cin.get(c)) {
        s += c;
    }
    s += '!';
    std::cout << s << std::endl;
    return 0;
}
```

ANSWER:

2.

```
int main() {
    std::vector<int> v(10);
    for(int i = 0; i < v.size(); ++i)
    v[i] = 5 - i;
    while(!v.empty()) {
        std::cout << v.back() << std::endl;
        v.pop_back();
    }
    return 0;
}
```

ANSWER:

3.
```
int main() {
    std::map<int,char> m;
    m[3] = 50;
    m[2] = 1;
    m[4] = 50;
    m[5] = 48;
    m[2] = 115;
    m[1] = 99;
    std::stringstream ss;
    for(std::map<int,char>::reverse_iterator it = m.rbegin();
        it != m.rend(); ++it) {
        ss << it->first << ": " << it->second << std::endl;
    }
    std::cout << ss.str();
    return 0;
}
```

ANSWER:

4.
```
float foo(int& x, int y) {
    x *= 1.5;
    y *= 0.5;
    return x + y;
}

int main() {
    int a = 2;
    float b = 3.f;
    int c = foo(a,b);
    std::cout << "a = " << a << ", b = " << b << ", foo(a,b) = "
        << c << std::endl;
    return 0;
}
```

ANSWER:

5.
```
class Foo {
    public:
    Foo() : str("\0") {}
    std::string getStr() const { return str; }
    private:
    std::string str;
};

int main() {
    Foo bar;
    std::cout << bar.getStr().length();
    return 0;
}
```

ANSWER:

Part II: Code Explanation. [2 points each problem]

Give an explanation, in complete English sentences, of the following concepts.

TIP

The explanation should be sufficient for a classmate to understand the concept, even if they had not previously been exposed to it.

6. Explain what is wrong with the following code fragment (other than missing `#include`s):

```
using namespace std;
using vit = vector<string>::iterator;
void print(const vector<string>& s) {
    for(vit it = s.begin(); it != s.end(); ++it ){
        cout << *it << endl;
    }
}
```

ANSWER:

7. Explain what is wrong with the following code fragment (other than missing *#include*s):

```
class Foo {
private:
    char label;
};

int main() {
    Foo bar;
    bar.label = 'c';
    return 0;
}
```

ANSWER:

8. Explain what is wrong with the following code fragment (other than missing *#include*s):

```
class Foo {
public:
    Foo(int v): _val(v) { }
    int getVal() const { return _val; }
private:
    int _val;
};

int main() {
    Foo bar;
    std::cout << "my foo's val is " << bar.getVal() << std::endl;
    return 0;
}
```

ANSWER:

Part III: Code Writing. [3 points each problem]

Write a code fragment to perform the specified task; remember, it's a fragment, so you don't have to show the *#include* statements or `main()` unless specifically requested, but you should write proper C++ code that could be compiled and run correctly.

9. In mathematics, a hyperbolic number is represented by two real numbers x and y, and is often denoted as $x + yj$, where j represents the non-real square root of 1. i.e. j^2=1, but j is not a real number. Consider the following file named `Hyperbolic.h` which includes a definition for a class named `Hyperbolic` to represent an arbitrary hyperbolic number:

```
// Hyperbolic.h
#ifndef HYPERBOLIC_H
#define HYPERBOLIC_H
class Hyperbolic {
private:
    double _x; // first real number x in "x + yj"
    double _y; // second real number y in "x + yj"
public:
    Hyperbolic(double x, double y):_x(x), _y(y) {}
    double getX() const { return _x; }
    double getY() const { return _y; }
    void setX(double x) { _x = x; }
    void setY(double y) { _y = y; }

    /* adds other to this number and returns a new complex number
    object representing its sum */
    Hyperbolic add(Hyperbolic& other) const;    //prototype only

    /* outputs to standard output the hyperbolic number in the form
    x + yj. you do not need to do any pretty printing e.g. if,
    for example, y is negative x + -yj is fine */
    void output() const;                        //prototype only
};
#endif
```

The procedure for adding hyperbolic numbers $x + yj$ and $u + vj$ is to add, respectively, the two real numbers, so that the sum of the two is the new number given by $(x + u) + (y + v)j$.

In the space below, show the complete contents of a file named `Hyperbolic.cpp` which would supply the definition of the add and output methods whose prototypes are given above marked by the comments. In this problem, show the entire contents of `Hyperbolic.cpp` (including all required `#include` and `using` statements that would need to appear in it).

ANSWER:

10. Define the function below which extends a given string into a string twice as long, by mirroring the characters of the original in place. If the given string s is initially ”abc”, the function changes the value of s to be ”abccba”

ANSWER:

```
void copy_chars_mirror(std::string& s) {













}
```

11. Complete the C++ function whose header is given below. The function should take as input a `std::vector` of `std:strings`, and return a `std::vector` of `std:strings` containing only values ***that appeared in the vector at least twice*** (in a case-insensitive manner).

    For example, if `vec` contains the values

    `dog, mouse, cat, tiger, mouse, Mouse, tiger, DOG`

    the vector returned by this method should contain:

    `dog, mouse, tiger`

    The values in the returned vector should be in ASCII order and contain only lower cases as shown above.

TIP

Aim to use the STL to your advantage here; appropriate usage can make your job easier!

ANSWER:

```
using namespace std;
vector<string> getRepeatedValuesVector(const vector<string>& vec) {















}
```

<The END of Homework 6>