Name: Felipe Lima

Complete these activities to the best of your abilities without using the internet or your computers. You will not be graded on correctness of your answers or on syntactic correctness. This exercise is for us and you to see where your c++ memory and experience currently is.

1 C++ General Review

1. What is a class?
A. A class is a date type that's holds its own information either public or private and its used to create objects and different instances of the same class.

2. What is a class attribute? (also called a "field")
A. A class attribute is variable or function member of the class.

3. What makes a class attribute different than a local variable?
A. A class attribute can be accessed by the functions in the class and stores information outside one specific function.

4. What does the keyword this refer to and why is it important?
A. This is a pointer to the current object instance. It can be used to refer to variables in a class from within a member function, especially if a variable has the same name in a local scenario.

5. What is a struct? How are classes and structs different from one another?
A. Struct is a group of variables under the same name. One big difference from classes is security. Structs are not secure while classes can hide its programing details.

6. What is a pointer? Explain, then give an example declaration of a pointer in c++.
A. A pointer is an object that stores the value of a memory address.
Int a = 10;
Int* ptr = &a;

7. Define a struct Node to be used in a linked list. Your linked list should contain integer data. You may choose between a singly linked and a doubly linked list.
A.

```
struct Node {
    int data;
    struct Node* next;
};
struct Node* head = NULL;
```

8. What are the differences between the following three variable declarations? Show how you would increase the value of x from the variables x, y, and z.

(a) int x = 2;
A. declared x as a variable of type int.
To increase value: x+= 10;

(b) int &y = x;
A. declares a reference called y for x; the address of y and x are the same, meaning altering one of them alters both
To increase value: y += 10;

(c) int *z = &x;
A. declares z as a pointer to x;
To increase value: *z += 10;

9. Recall that std::cout in the iostream library is used to print text to standard out. Write a c++ hello world program.
A.

```
int main() {
    std::cout << "Hello, World! \n";

  return 0;
}
```

10. Write a command that you would use to compile your hello world program from the command line, then write the command to run the executable that your compilation command produces.
A.
g++ -std=c++11 test.cpp -o test
./test

2 Code Comprehension

```
1  #include <iostream>
2
3  struct Thing {
4      int a;
5      double b;
6  };
7
8  int main() {
9      Thing t1;
10     std::cout << t1.a << std::endl;
11 }
```

(a) Is there a compile time error for the given code snippet?
A. No

(b) Is there a runtime error for the given code snippet?
A. trying to access t1.a without a value, should just present some garbage value.

(c) If there are no errors, what do you think the output would be? Would it be the same each time you ran the program?
A. The output should be just some garbage value different each time you run the program.

```cpp
#include <iostream>


int main(int argc, char* argv[]) {
    std::cout << argv[0] << std::endl;
    std::cout << argv[1] << std::endl;
}
```

(a) This code snippet is contained in the file ex2.cpp was compiled using the command *g++ -std=c++11 -Wall ex2.cpp -o ex2*. Write two commands to run the produced executable from the command line; one that does not produce an error and one that does produce an error.

A.

No error: ./ex2 hello

Error: ./ex2

(b) Modify the given code so that neither of the commands given in the previous question would produce errors.

A.

Delete the line: std::cout << argv[1] << std::endl;

```cpp
#include <iostream>

void Swap1(int x, int y) {
    int tmp = x;
    x = y;
    y = tmp;
}

void Swap2(int &x, int &y) {
    int tmp = x;
    x = y;
    y = tmp;
}

int main(int argc, char **argv) {
    int a = 5;
    int b = 10;

    Swap1(a, b);
    std::cout << "a: " << a << "; b: " << b << std::endl;

    a = 5;
    b = 10;
    Swap2(a, b);
    std::cout << "a: " << a << "; b: " << b << std::endl;
    return EXIT_SUCCESS;
}
```

What is the output of the given code? Explain why the output is what you wrote down.
A.
a: 5; b: 10
a: 10; b: 5

Swap1 doesn't swap the variables a and b, only x and y on the scope of the function

Swap2 swaps the variable because it was passed into the function by reference.

```cpp
#include <iostream>

class Person {
public:
    std::string name;
    int age;

    void IncreaseAge() { age++;};
};

int main() {
    Person p1;
    p1.name = "Alice";
    p1.age = 10;

    // section 1
    std::cout << p1.age << std::endl;
    p1.IncreaseAge();
    std::cout << p1.age << std::endl;

    Person * p2 = &p1;
    // your function call to IncreaseAge via the p2 variable here


    // section 2
    std::cout << p1.age << std::endl;
    std::cout << p2->age << std::endl;

    Person * p3 = new Person();
    // your function call to IncreaseAge via the p3 variable here


    // section 3
    std::cout << p1.age << std::endl;
    std::cout << p2->age << std::endl;
    std::cout << p3->age << std::endl;
}
```

(a) What is the output of section 1 of the code? (the first two couts)
A.
10
11

(b) Add a function call to IncreaseAge via the p2 variable to the indicated part of the code
A.
p2->IncreaseAge();

(c) What is the output of section 2 of the code?
A.
12
12

(d) Add a function call to IncreaseAge via the p3 variable to the indicated part of the code
A.
P3->IncreaseAge();

(e) What is the output of section 3 of the code?
A.
12
12
1

```
#include <iostream>
#include <cstdlib>
#include <math.h>

void Mystery(int x) {
    int u = (int)sqrt((double)x);
    bool *a = new bool[x + 1];

    // sets all entries in a to 0 (false)
    memset(a, 0, sizeof(bool) * (x + 1));

    // loop 1
    for (int m = 2; m <= u; m++) {
        if (!a[m]) {
            for (int k = m * m; k <= x; k += m) {
                a[k] = true;
            }
        }
    }

    // loop 2
    for (int m = 2; m <= x; m++) {
        if (!a[m]) {
            std::cout << m << " ";
        }
    }

    delete [] a;
}


int main(int argc, char* argv[]) {
    Mystery(atoi(argv[1]));
    return EXIT_SUCCESS;
}
```

(a) What type is a?
A.
Pointer of type bool

(b) How many elements does the array hold in relation to x?
A.
One more than x

(c) If x is 10, what is the value of u?
A.
3

(d) If x is 10, what is the value of a after the end of loop 1?
A.
1

(e) What is the purpose of loop 2?
A.
To print “m “ as many times as x-2

(f) What does the Mystery function do?
A.
I don’t really know how to explain.

End of Worksheet