CMPSC 32 F18
Object Oriented Design and Implementation

Midterm 1 Examination

Please state your answers as clearly as possible. This exam not only tests your understanding of the material, but also how well you can convey your understanding to us. Remember that you are solely responsible for the answers to the questions, therefore, please refrain from consulting with your class peers.

Please write all your answers **LEGIBLY** and **CLEARLY**. If we cannot decipher your answers, you will not receive credit.

No electronic devices are allowed during the exam (calculators, cell phones, laptops, etc.).

READ all questions carefully before attempting to answer. If there are any ambiguities in the statement of questions, please ask us. **You may assume that each problem is correct and solvable unless the question specifically asks about errors.**

THE GRADE IN THIS EXAM IS A TOTAL OF 39 POINTS.

Name (as it would appear on the official course roster)	**Umail Address**
	@umail.ucsb.edu

Question 1 (6 points)

Write whether each statement is True or False. If False, briefly state why (1 point each)

a. C++ will always provide a default constructor for a class if one is not defined.

b. C++ will always provide a default copy constructor if one is not defined.

c. `std::map` is implemented using a hash table structure.

d. Regardless of the input values, selection sort will run in $O(n^2)$ in all cases.

e. Unless specified, a struct's members are automatically defined to have private accessibility.

f. An `std::set` container may have duplicate values.

Question 2 (5 points)

a. Given the following variables `CXX` and `DEPENDENCIES` in a Makefile, write a Makefile target called `app` that will use the dependencies that will generate an executable named `app` using the C++11 standard. You **must** use the `$@` and `$^` special characters in your solution, as well as the given variables where appropriate.

```
CXX=g++
DEPENDENCIES=X.o Y.o Z.o

app:
```

b. Assuming no object (.o) files or executable file (`app`) exists when running the make command from part a, state all the files that will be generated after the command above finishes execution. You may assume the appropriate source files (.cpp) exist.

Question 3 (4 points)

Consider the following struct definition:

```
class A {
public:
	int w; short x; int y; short z;
};
```

Write the output in the blank for the following code snippet:

```
A a;
cout << &a << endl;		// given output: 0xf17548

cout << &a.w << endl;	_____________________________

cout << &a.x << endl;	_____________________________

cout << &a.y << endl;	_____________________________

cout << &a.z << endl;	_____________________________
```

Question 4 (4 points)

Given the following namespace definitions and main function, write the output in the blank space to the right of each statement. If the code does not compile, simply write ERROR. You may treat each line independently from each other.

```
using namespace std;
namespace A { void f() { cout << "A.f()" << endl; } }
namespace B { void f() { cout << "B.f()" << endl; } }
void f() { cout << "f()" << endl; }
using namespace A;

int main() {
	f();		_____________________________

	A::f();		_____________________________

	B::f();		_____________________________

	::f();		_____________________________
}
```

Question 5 (8 points)

a. **Briefly** describe the difference between a **shallow copy** and a **deep copy**.

b. **Briefly** explain why vectors containing different number of elements all have the same `sizeof()` result.

c. Briefly state what the optimization of the bubblesort algorithm discussed in class is. What is the best-case running time for bubblesort assuming this optimization is implemented.

Question 6 (5 points)

Given an incomplete definition of insertionSort, fill in the blanks with a constant, variable, or expression such that the function will work correctly.

```
void insertionSort(int a[], size_t size) {
	int item;
	int shiftIndex;
	for (int i = _____________________; i < _____________________; i++) {

		item = a[_____________________];

		shiftIndex = _____________________;

		while (_____________________ >= 0 && a[_____________________] > item) {

			a[_____________________] = a[_____________________];
			shiftIndex -= 1;
		}

		a[_____________________] = _____________________;
	}
}
```

Question 7 (7 points)
Given a class interface (.h file) containing the default and overloaded constructor definitions for an object X, implement the "Big Three" (copy constructor, assignment operator, and destructor) for the class X in a corresponding .cpp file. Your implementation should avoid any memory leaks and provide deep copies for all the class member variables.

```
// X.h
#ifndef X_H
#define X_H

class X {
public:
	X() {
		this->a = new int(0);
		this->b = new double(0);
	}
	X(int a, double b) {
		this->a = new int(a);
		this->b = new double(b);
	}
	X(X &obj);                          // implement copy constructor
	~X();                               // implement destructor
	X& operator=(const X& rhs);         // implement overloaded assignment operator
	int* getA() { return a; }
	double* getB() { return b; }
private:
	int* a;
	double* b;
};

#endif
-----
// X.cpp

// Define the "Big Three" below. You may assume all necessary libraries are already included
// in your implementation and only need to provide the function definitions (including the
// function signatures) in this file.
```

Scratch / Question paper