CMPSC 32 S18
Object Oriented Design and Implementation

Midterm 2 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 50 POINTS.

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

Question 1 (8 points)

For the following segment of code, write a ✓ next to the line of code if that line of code will produce **NO** error. Write an X next to the line of code if that line of code **WILL** produce an error.

```
class A { virtual void X() = 0; };
class B : public A { virtual void X() {} };
class C : public A { virtual void X() {} };
class D : public C { virtual void X() {} };


A* a1 = new A(); ______                    C* c1 = new A(); ______

A* a2 = new B(); ______                    C* c2 = new B(); ______

A* a3 = new C(); ______                    C* c3 = new C(); ______

A* a4 = new D(); ______                    C* c4 = new D(); ______

B* b1 = new A(); ______                    D* d1 = new A(); ______

B* b2 = new B(); ______                    D* d2 = new B(); ______

B* b3 = new C(); ______                    D* d3 = new C(); ______

B* b4 = new D(); ______                    D* d4 = new D(); ______
```

Question 2 (12 points)

a. For the following code, write the resulting output.

```
class A {
public:
	A() {}
	~A() { std::cout << "A::~A()" << std::endl; }
	void function1() { std::cout << "A::function1()" << std::endl; }
	virtual void function2() { std::cout << "A::function2()" << std::endl; }
};
class B : public A {
public:
	B() {}
	void function1() { std::cout << "B::function1()" << std::endl; }
	virtual void function2() { std::cout << "B::function2()" << std::endl; }
	virtual void function3() = 0;
};
class C : public B {
public:
	C() {}
	~C() { std::cout << "C::~C()" << std::endl; }
	void function1() { std::cout << "C::function1()" << std::endl; }
	virtual void function2() { std::cout << "C::function2()" << std::endl; }
	virtual void function3() { std::cout << "C::function3()" << std::endl; }
	void function4() { std::cout << "C::function4()" << std::endl; }
};
void function1() {
	C c1;
	A a1 = c1;
	a1.function1();
	a1.function2();
	B* b1 = new C();
	b1->function1();
	b1->function2();
	b1->function3();
}
int main() {
	function1();
	return 0;
}
```

b. Explain what memory slicing is and provide a brief example of memory slicing using the classes defined above.

Question 3 (9 points)
For the following code, write the resulting output:
```
class A {};
class B {};
class C : public B {};
class D : public A {};
int main() {
       int input[] = {1,2,3,4};
       for (int i = 0; i < 4; i++) {
              int x = input[i];
              try {
                     if (x % 3 == 0) { throw A(); }
                     if (x % 2 == 1) { throw B(); }
                     if (x % 3 == 1) { throw C(); }
                     if (x % 3 == 2) { throw D(); }
              }
              catch (D e) { std::cout << "Caught D" << std::endl; }
              catch (A e) { std::cout << "Caught A" << std::endl; }
              catch (B e) { std::cout << "Caught B" << std::endl; }
              catch (C e) { std::cout << "Caught C" << std::endl; }
              std::cout << "End Loop" << std::endl;
       }
       std::cout << "Done" << std::endl;
}
```

Question 4 (7 points)
Write whether each statement is True or False. If False, briefly state why (1 point each)

a. A copy of an array is made when it is passed to a function.

b. The Standard Template Library `map` data structure is inherently in sorted order.

c. The `virtual` keyword is necessary to allow subclasses to inherit functions from its base class.

d. *Double Hashing* is a technique used to handle collisions by placing the item in the next open array slot.

e. When testing, *Error Cases* test the edge values of valid inputs.

f. *Protected Inheritance* is when public and protected members of a base class become private members of a derived class.

g. A *virtual function* is **required** to be implemented by a derived class.

Question 5 (14 points)

a. What are the best, average, and worst-case running times of quicksort and mergesort? **Briefly** compare the memory utilization cost for each algorithm (as implemented in lecture).

b. **Briefly** state the Test-Driven Development process.

c. **Briefly** describe what two problems with *open-address* hashing are.

d. **Briefly** explain why it is good practice (and very important) to declare base class destructors as virtual.

Scratch Paper (Do not detach)