2143 - OOP
Spring 2021
Take Home Exam
April 22, 2021

Name: Loic Konan

READ THESE INSTRUCTIONS
D Create a digital document (PDF) that has zero handwriting on it. Print and bring to the final exam on *Tuesday April 27th from 8:00 am - 10:00am.*
D Your presentation and thoroughness of answers is a large part of your grade. Presentation means use examples when you can, graphics or images, and organize your answers!
D I make every effort to create clear and understandable questions. You should do the same with your answers.
D Questions should be answered in order and clearly marked.
D Your name should be on each page, in the heading if possible.
D Place your PDF on GitHub (after you take the actual final) and in your assignments folder.
D Create a folder called **Take Home Exam** and place your document in there. Name the actual document: **exam.pdf** within the folder.
Failure to comply with any of these rules will result in a NO grade. This is a courtesy exam to help you solidify your grade.

Grade Table (do not write on it)

Question	Points	Score
1	70	
2	15	
3	40	
4	10	
5	15	
6	10	
7	10	
8	20	
9	15	
10	20	
11	35	
12	10	
13	10	
Total:	280	

Warning: Support each answer with details. I do not care how mundane the question is ... justify your answer. Even for a question as innocuous or simple as" What is your name?", you should be very thorough when answering:

what is your name? My name is Attila. This comes from the ancient figure" Attila the Hun". He was the leader of a tribal empire consisting of Huns, Ostrogoths, Alans and Bulgars, amongst others, in Central and Eastern Europe. My namesake almost conquered western Europe, but his brother died, and he decided to go home. Lucky for us! We would all be speaking a mix of Asiatic dialects :)

Single word answers, and in fact single sentence answers will be scored with a zero. This is a take-home exam to help study for the final and boost your grade. Work on it accordingly.

1. This VS That. Not so simple answers Ç:

   (a) (7 points) Explain the difference between a *struct* and a *class*. Can one do whatever the other does?

   (b) (7 points) What is the difference between a *class* and an *object*?

   (c) (7 points) What is the difference between *inheritance* and *composition*? Which one should you lean towards when designing your solution to a problem?

   (d) (7 points) What is the difference between a *deep* vs a *shallow* copy? What can you do to make one or the other happen?

   (e) (7 points) What is the difference between a *constructor* and a *destructor*? Are they both mandatory or even necessary?

   (f) (7 points) What is *static* vs *dynamic* typing? Which does C++ employ and which does Python employ?

   (g) (7 points) What is *encapsulation* vs *abstraction*? Please give some examples!

(h) (7 points) What is the difference between an *abstract class* and an *interface*?

(i) (7 points) What is the difference between a *virtual function* and a *pure virtual function*?

(j) (7 points) What is the difference between *Function Overloading* and *Function Overriding*?

2. Define the following and give examples of each:

(a) (5 points) Polymorphism

(b) (5 points) Encapsulation

(c) (5 points) Abstraction

3.

(a) (5 points) What is a default constructor?

(b) (5 points) What is an overloaded constructor? And is there a limit to the number of overloaded constructors you can have?

(c) (5 points) What is a copy constructor? Do you need to create a copy constructor for every class you define?

(d) (5 points) What is a deep copy, and when do you need to worry about it?

(e) (5 points) Is there a relationship between copy constructors and deep copying?

(f) (5 points) Is a copy constructor the same as overloading the assignment operator?

(g) (10 points) Give one or more reason(s) why a class would need a destructor.

4. (10 points) What is the difference between an abstract class and an interface?

Hint:
You should include in your discussion:

- Virtual Functions
- Pure Virtual Functions

5. Describe the following (make sure you compare and contrast as well):

(a) (5 points) Public -

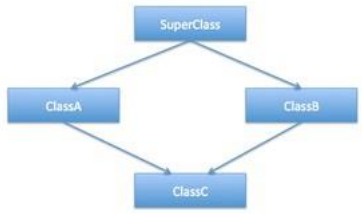
(b) (5 points) Private –

(c) (5 points) Protected -

Hint:

- Make sure you define each item individually as well.
- Use examples.
- If you're not sure, use examples to make your point.
- Ummm, example code is always welcome.

6. (10 points) What is the diamond problem?



Hint:

- This is a question about multiple inheritance and its potential problems.
- Use examples when possible but explain thoroughly.

7. (10 points) Discuss Early and Late binding.

Hint:

- These keywords should be in your answer: **static, dynamic, virtual, abstract, interface**.
- If you have not figured it out .... use examples.

8. (20 points) Using a **single** variable, execute the show method in *Base* and in *Derived*. Of course you can use other statements as well, but only one variable.

```
class Base{
        public:
        virtual void show() { cout <<" In Base n"; }
};

class Derived: public Base{
        public:
        void show() { cout <<"In Derived n"; }
};
```

Hint: This is implying that dynamic binding should be used. A pointer to the base class can be used to point to the derived as well.

9. (15 points) Given the two class definitions below:

```
class Engine {} // The Engine class.

class Automobile {} // Automobile class which is parent to Car class.
```

You need to write a definition for a Car class using the above two classes. You need to extend one and use the other as a data member. This question boils down to composition vs inheritance. Explain your reasoning after you write your Car definition (bare bones definition).

10. (20 points) Write a class that contains two class data members *numBorn* and *numLiving*. The value of *numBorn* should be equal to the number of objects of the class that have been instanced. The value of *numLiving* should be equal to the total number of objects in existence currently (i.e., the objects that have been constructed but not yet destructed.)

11.

(a) (10 points) Write a program that has an abstract base class named *Quad*. This class should have four member data variables representing side lengths and a *pure virtual function* called *Area*. It should also have methods for setting the data variables.

(b) (15 points) Derive a class *Rectangle* from *Quad* and override the *Area* method so that it returns the area of the Rectangle. Write a main function that creates a Rectangle and sets the side lengths.

(c) (10 points) Write a top-level function that will take a parameter of type *Quad* and return the value of the appropriate Area function.

Note: A **top-level function** is a function that is basically stand-alone. This means that they are functions you call directly,

without the need to create any object or call any class.

12. (10 points) What is the rule of three? You will have answered this question (in pieces) already, but in the OOP world, what does it mean?

13. (10 points) What are the limitations of OOP?