2143 - OOP
Spring 2021
Take Home Exam
April 25, 2021

Name: Seth Allen

READ THESE INSTRUCTIONS
› 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.*
› Your presentation and thoroughness of answers is a large part of your grade. Presentation means: use examples when you can, graphics orimages, and organize your answers!
› I make every effort to create clear and understandable questions. You should do the same with your answers. › Questions should be answered in order and clearly marked.
› Your name should be on each page, in the heading if possible.
› Place your PDF on GitHub (after you take the actual final) and in your assignments folder.
› Create a folder called **TakeHomeExam** 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 (don't 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 and every 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 studyfor the final and boost your grade. Work on it accordingly.

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

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

Structs are value type whereas Classes are reference type. Structs are stored on the stack whereas Classes are stored on the heap.

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

A class is a template for an object an object is a member or instance of that class. An object has a state in which all of its properties have values that either explicitly define or that are defined by default settings.

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

Inheritance derives one class from another, Composition however defines a class as the sum of its parts.

(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?

Shallow copy constructs a new compound object and then inserts references into it found in the original. A deep copy constructs a new compound object and then, recursively, inserts copies into it from the original objects.

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

Constructor is used to initialize the object. Whereas destructors are used to destroy the instances. C++

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

Static typing language are those in which type checking is done at compile-time, whereas dynamic typed language is typed checked at run-time. C++ employs Static typed Python employs 2 types strongly typed and dynamically typed. Strong typing means that variables do have a type and that the type matters when performing operations on a variable.

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

Abstraction is the method of hiding unwanted information. Encapsulation is a method to hide the data in a single entity or unit along with a method to protect information from outside.
Encapsulation can be implemented by access modifier like public, private, and protected
Abstraction examples would be like ATM operations or Vehicle operations.
Encapsulation examples like capsules to encapsulate several different kinds of medicine.

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

An abstract class allows you to create functionality that subclasses can implement or override. An interface only allows you to define functionality, not implement it.

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

A virtual function is a member function of base class which can be redefined by derived class.

A pure virtual function is a member function of base class whose only declaration is provided in base class and should be defined in derived class.

(j) (7 points) What is the difference between *Function Overloading* and *Function Overriding*?
Function overloading is when multiple functions with the same name exist in a class.
Function overriding is when function have same prototype in base class as well as derived class.

2. Define the following and give examples of each:

(a) (5 points) Polymorphism: Allows programmers to perform a single task in numerous ways. Example: Would be a class called animal and a method called sound since its in a generic class it has to have a generic message

(b) (5 points) Encapsulation: bundles data along with the methods that operate on that data into a single unit
The best example would be a calculator.

(c) (5 points) Abstraction: this only shows essential attributes and hides unnecessary information
Abstractions example would be a coffee machine you supply it with the necessary ingredients and switch it on what you don't see is the internal components of that coffee machine.

3. (a) (5 points) What is a default constructor?
Constructors are functions of a class that are executed when new objects of the class are created. They are used when the programmer doesn't define a variable

(b) (5 points) What is an overloaded constructor? And is there a limit to the number of overloadedconstructors you can have?
The constructor overloading can be defined as the concept of having more than one constructor with different parameters so that every constructor can perform a different task. No there is no limit to overloaded constructors.

(c) (5 points) What is a copy constructor? Do you need to create a copy constructor for every class you define?
A copy constructor is a member function that initializes an object using another object of the same class. No you do not need 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?
Deep copy is a process in which the copying process occurs recursively. It means first constructing a new collection object and then recursively populating it with copies of the child objects found in the original. In case of deep copy, a copy of object is copied in other object.

(e) (5 points) Is there a relationship between copy constructors and deep copying?
no deep copy copies all fields into a new object while just copying points to the original object.

(f) (5 points) Is a copy constructor the same as overloading the assignment operator?
Both are used to initialize one object into another object. however they are separate at the point of how they initialize objects copy creates separate memory location but the assignment operator uses one memory location.

(g) (10 points) Give one or more reason(s) why a class would need a destructor.
Destructors are usually used to deallocate memory and do other cleanup for a class object and its class members when the object is destroyed.

---

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

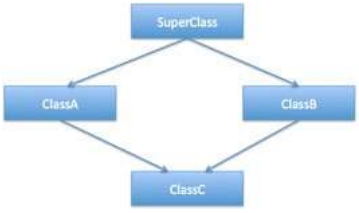
---

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

(a) (5 points) Public: methods specified as public are accessible from outside of the class
(b) (5 points) Private: methods specified as private can only be accessed by the functions inside the class
(c) (5 points) Protected: methods declared as protected can only be accessed by subclasses. unlike private or public where private methods can only be accessed by that class

Hint:
- Make sure you define each item individually as well.
- Use examples.
- If your 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.

The diamond problem is an ambiguity that arises when two classes b and c inherit from a and class d inherits from both B and C.

---

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

Hint:

- These keywords should be in your answer: **static, dynamic, virtual, abstract, interface**.
- If you haven't 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.

Base *ptr;
ptr->show();
Derived ptr;
ptr=&ptr;
ptr ->show();

---

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 your write you Car definition (bare bones definition).

class Car
{
        public int Engine();
        public int Automobile():
};

10. (20 points) Write a class that contains two class data members *numBorn* and *numLiving*. The valueof *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 objectsthat have been constructed but not yet destructed.)

class life

{

        private:

```
        int numBorn = 2;
        int numLiving = 2;
};
```

---

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.

```
class Quad
{
    lengthR = lengthL = SideT = SideB = 0;
};
```

(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.

```
class Rectangle: private Quad
{
    Rectangle()
    {
        lengthR= lengthL = 24;
        sideB= SideT = 12;
    }
};
```

(c) (10 points) Write a top-level function that will take a parameter of type *Quad* and return thevalue 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 ca 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, butin the OOP world, what does it mean?
There are 3 techniques Inheritance, Encapsulation, Polymorphism.

---

13. (10 points) What are the limitations of OOP?
size, effort, and speed OOP program are much larger than other programs, they require a lot of work to create. OOP programs are slower than other programs partially because of their size

Citations

"Constructor Overloading in Java - Javatpoint." *Www.Javatpoint.Com*, www.javatpoint.com/constructor-overloading-in-java. Accessed 29 Apr. 2021.

"Copy — Shallow and Deep Copy Operations — Python 3.9.4 Documentation." *Python*, docs.python.org/3/library/copy.html. Accessed 29 Apr. 2021.

"Default Constructors in C++." *Tutorialspoint*, www.tutorialspoint.com/default-constructors-in-cplusplus. Accessed 29 Apr. 2021.

GeeksforGeeks. "Difference between Virtual Function and Pure Virtual Function in C++." *GeeksforGeeks*, 9 June 2020, www.geeksforgeeks.org/difference-between-virtual-function-and-pure-virtual-function-in-c.

Kanjilal, Joydip. "When to Use an Abstract Class vs. Interface in C#." *InfoWorld*, 1 Jan. 2018, www.infoworld.com/article/2928719/when-to-use-an-abstract-class-vs-interface-in-csharp.html#:%7E:text=The%20short%20answer%3A%20An%20abstract,take%20advantage%20of%20multiple%20interfaces.

Libretexts. "A.1: Difference between Abstraction and Encapsulation in C++." *Engineering LibreTexts*, 25 Sept. 2020, eng.libretexts.org/Courses/Delta_College/C_-_Data_Structures/06%3A_Abstraction_Encapsulation/1.01%3A_Difference_between_Abstraction_and_Encapsulation#:%7E:text=Abstraction%20is%20the%20method%20of,to%20protect%20information%20from%20outside.&text=Whereas%20encapsulation%20can%20be%20implemented,i.e.%20private%2C%20pr

otected%20and%20public.

Singh, Chaitanya. “Polymorphism in Java with Example.” *Beginnersbook.Com*, 19 Aug. 2018, beginnersbook.com/2013/03/polymorphism-in-java.

“Understanding Classes and Objects.” *NCAR Command Language*, www.ncl.ucar.edu/Document/HLUs/User_Guide/classes/classoview.shtml#:%7E:text=The%20difference%20is%20simple%20and,is%20a%20template%20for%20objects.&text=An%20object%20is%20a%20member,are%20defined%20by%20default%20settings. Accessed 29 Apr. 2021.

“What’s the Difference between Struct and Class in .NET?” *Stack Overflow*, 16 Aug. 2008, stackoverflow.com/questions/13049/whats-the-difference-between-struct-and-class-in-net#:%7E:text=Difference%20between%20Structs%20and%20Classes,are%20stored%20on%20the%20heap.&text=When%20you%20copy%20struct%20into,value%20of%20the%20other%20struct.

Wikipedia contributors. “Object Copying.” *Wikipedia*, 17 Apr. 2021, en.wikipedia.org/wiki/Object_copying.