2143 - OOP **Name:** _____
Fall 2020
Take Home Exam
November 25, 2020

READ THESE INSTRUCTIONS

- What to turn in: a PDF with your name and NO handwriting.
- Cut and paste the questions below into Word or Google Docs (or whatever you use) and create a document that can be saved as a PDF. Your name should be on each page, in the heading if possible.
- Answer Sheet Starter: See our slack channel.
- Your presentation and thoroughness of answers is part of your grade. I make every effort to create clear and understandable questions. You should do the same with your answers.
- Your answer sheet should be clearly marked with answers in order.
- Turn your final in by uploading to D2L. Link should be obvious on D2L course main page.
- **1 or 2 questions will deal with topics that we didn't to cover in class, but I still feel you should know about. Take home exams! Yes!**
- **Questions that sound like a 'yes' or 'no' will suffice, is never true! Always explain and use examples whenever possible.**

Failure to comply will result in loss of letter grade

Grade Table (don't write on it)

Question	Points	Score
1	10	
2	15	
3	15	
4	10	
5	15	
6	10	
7	10	
8	20	
9	20	
10	20	
11	20	
12	10	
13	20	
14	15	
15	15	
Total:	225	

1. (10 points) What is the main difference between a class and an object?.

---

2. (15 points) Define the following and give examples of each :
   - **(5 points)** Polymorphism
   - **(5 points)** Encapsulation
   - **(5 points)** Abstraction

---

3. (15 points) Discuss constructors for objects. In your discussion hit on these points:
   - **(5 points)** Are they always necessary? If so why.
   - **(5 points)** What is a copy constructor? Is it the same as overloading the assignment operator?
   - **(5 points)** Deep vs Shallow copy? Give examples when each are needed.

---

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

Note:

You should include in your discussion:
- Virtual Functions
- Pure Virtual Functions

---

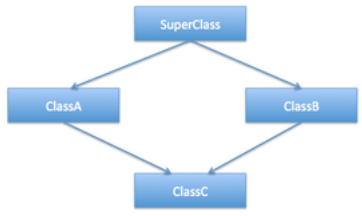
5. (15 points) What are the differences between:
   - **(5 points)** Public
   - **(5 points)** Private
   - **(5 points)** Protected

Note:

Again, use examples if possible. Make sure you define each item individually as well.

---

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



Note:

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.

Note:

You should have these keywords present when answering this question: **static, dynamic, virtual, abstract, interface**. I can't say it enough, try to use examples in your discussion.

---

8. (20 points) What is a design pattern? Describe the following patterns and give examples of when to use:

   - **(5 points)** Facade
   - **(5 points)** Singleton
   - **(5 points)** Factory
   - **(5 points)** Observer

Note:

I used the term "design pattern" when we talked about "static" data members. Remember the "borg" pattern? We by no means talked in depth about them, or are they in the course notes. But, they are important. Look up the above design patters (google) and answer the question.

---

9. (20 points) Write a class called `NumObjects` that counts the number of objects in existence. You should assume that each object will be created and destroyed before your program ends. So your count should be equal to the number of existing objects.

10. (20 points) Given the code below. Update it so that both Rectangle AND Triangle are forced to overload the area method.

```
class Polygon {
protected:
    double width, height;

public:
    void set_values(double a, double b) {
        width = a;
        height = b;
    }
};

class Rectangle : public Polygon {
public:
    double area() {
        return width * height;
    }
};

class Triangle : public Polygon {
public:
    double area() {
        return width * height / 2;
    }
};
```

---

11. (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"; }
};
```

---

12. (10 points) Will this snippet run? If not explain why.

```
class Base{
public:
    virtual void show() = 0;
};

int main(void){
    Base b;
    Base *bp;
    return 0;
}
```

---

13. (20 points) There is a class below called `GreatCircleGeo` that does distance calculations using the 'haversine' formula. You have a class called `GeodisicGeo` that has its own 'Geodisic' Distance function using the 'vincenty' formula. The problem is they each want to access each others distance function so they can use the correct distance function in certain situations. Can you rewrite one or both of the classes so that they can access each others distance functions?

```
class GreatCircleGeo{
    double d;

    double Haversine(point a, point b){
        double d = haversine_distance(a,b);
        return d;
    }

public:
    double Distance(point a, point b){ return Haversine(point a, point b)}
};

class GeodisicGeo{
    double d;

    double Geodisic(point a, point b){
        double d = vincenty_distance(a,b);
        return d;
    }

public:
    double Distance(point a, point b){ return Geodisic(point a, point b)}
};
```

14. (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).

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

```
class Thingy {
public:
    static int thingCount;

    // Constructor definition
    Thingy(string n) {
        name = b

            // Increase every time object is created
            thingCount++;
    }
    double GetName() {
        return name;
    }

private:
    string name;
};
```

Write a member function for Thingy, that would let someone get the value in thingCount.