Name: [redacted] *(as it would appear on official course roster)*	
Umail address: [redacted] @umail.ucsb.edu	section [redacted]
Optional: name you wish to be called if different from name above.	
Optional: name of "homework buddy" (leaving this blank signifies "I worked alone"	

1

h03

CS24 W19

h03: Chapter 3, section 3.1, Chapter 4, 4.1 - 4.5

ready?	assigned	due	points
true	Wed 01/23 02:00PM	Mon 01/28 09:00AM	100

You may collaborate on this homework with AT MOST one person, an optional "homework buddy".

MAY ONLY BE TURNED IN IN THE LECTURE/LAB LISTED ABOVE AS THE DUE DATE, OR IF APPLICABLE, SUBMITTED ON GRADESCOPE. There is NO MAKEUP for missed assignments; in place of that, we drop the lowest scores (if you have zeros, those are the lowest scores.)

Complete your reading of Chapter 3, section 3.1, Chapter 4, 4.1 - 4.5 (If you don't have a copy of the textbook yet, there is one on reserve at the library under "COMP000-STAFF - Permanent Reserve").

Please:

- **No Staples.**
- **No Paperclips.**
- **No folded down corners.**

1. (10 pts) What is the output of the following code:

```
int *p, *q;
int x;
p = &x;
q = new int;
*p = 50;
*q = 60;
cout << *p << ", "   << *q << endl;
p = q;
cout << *p << ", "  <<  *q << endl;
*p = 30;
*q = 40;
cout << *p << ", " << *q << endl;
delete p;
```

p q x new 50 60

50, 60

60, 60

40, 40

2. (5 pts) Suppose that we added the line `delete q;` after the last line of the code in the previous question, how would the behavior of the program change?

most likely a segfault since one memory location is freed up twice.

3. (10 pts) On page 98 (Chapter 3), the author talks about value semantics, which specifies different ways of copying the value of objects of a class to other objects of the same class. Using examples write two ways in which this can be done for the point class covered in lecture. You can find the code for that class here: https://github.com/ucsb-cs24-w19-mirza/cs24-w19-lectures/tree/master/lec-04

Option 1: point p(p1.get_x(), p1.get_y());

Option 2: point p; p.shift(p1.get_x(), p1.get_y());

Also, Option 3: point p; p = p1;

4. Read the code provided in pointCloud.h in point.h in this repo that describes a point and a pointCloud: https://github.com/ucsb-cs24-w18/cs24-h03. Answer the following questions about the provided code i. (5 pts) Suppose the parametrized constructor was implemented as below, what is the syntax/logic error. Provide a correct implementation.

2

h03

CS24 W19

ii. (10 pts) Implement the overloaded copy constructor

```
pointCloud::pointCloud(const pointCloud& other){
    this->numPoints = other.numPoints;
    this->capacity = other.capacity;
    this->cloud = new point[capacity];
    for(int i=0; i<numPoints; i++){
        *(this->cloud + i) = other->cloud[i];
    }
```

iii. (20 pts) Implement the overloaded assignment operator

```
void pointCloud::operator=(const pointCloud& other){
    delete [] this->cloud;
    this->numPoints = other.numPoints;
    this->capacity = other.capacity;
    this->cloud = new point[capacity];
    for(int i=0; i<numPoints; i++) this->cloud[i] = other->cloud[i];
}
```

iv. (20 pts) Implement the destructor

```
pointCloud::~pointCloud(){
    delete [] this->cloud;
}
```

v. (20 pts) Write (a), (b), (c), (d) or "none" against each of the following statements to indicate which of the following is invoked in each ca
(a) parameterized constructor, (b) copy constructor, (c) assignment operator, (d) destructor.

```
void whatsThePoint(){
        pointCloud *p = new pointCloud;          -> (a)
        for (int i = 0;i < 20; i++){
                p->insert(point(i, 10*i));       -> (a)
        }
        pointCloud q(*p);                        (b)
        pointCloud w;                            (a)
        w = q;                                   (c)
        delete p;                                (d)
}
```

```
pointCloud::pointCloud(int cap = 10){
        cloud = new point[capacity];
}
```

Syntax error: member var capacity is not initialized

logic error: member variable numPoints is not initialized

Correct implementation:

```
pointCloud::pointCloud(int cap=10){
    capacity = cap;
    numPoints = 0;
    cloud = new point[capacity];
}
```