Exercise #3 – Dynamic memory allocation, member initialization lists, static data members and functions.

Handed out: 10/1/15	Due: 10/6/15

Questions

1. [10 pts] How many bytes are required to represent an object of the class `Foo` for each of the definitions below?

```
struct Foo {
  char a;
  double b;
  int c;
};
```

```
struct Foo {
  char a;
  int b;
  double c;
};
```

2. [5 pts] How many bytes are required to represent an object of the class `Bar` defined below:

```
struct Bar {
  char a;
  int b;
  double c;
  static size_t d;
};
```

3. [5 pts] Given the following `new` expression, how would you delete `pa`?

```
int *pa = new int[10];
```

4. [5 pts] Assume that the class `MyClass` has a default constructor. Are the following legal and if so, what would it do? If not, why not?

```
MyClass x = new MyClass;
```

```
MyClass *x = new MyClass[10];
```

5. [5 pts] In the code below, insert a default constructor that initializes data member of class Foo to 0 using a member initialization list.

```
class Foo {
  private:
     const int data;
  public:
     // insert your code

};
```

6. [10 pts] Assume that the class `MyObject` has self-reporting versions defined for the default constructor, the constructor that takes a single integer argument, the assignment operator, and the destructor. What functions, and in what order, will be called when the function `foo()` is called:

```
void foo() {
     MyObject x(1);
     MyObject y = x;
     MyObject z(2);
     y = z;
     MyObject *w = new MyObject [10];
}
```

7. [10 pts] There are two problems with the following code using the class `Record` as defined below. What are they and how would you fix them?

Class definition:

```
class Record
{
private:
  const static int count = 1;
  int ID;
public:
  Record() : ID(count) { count++; }
  int getID() { return ID; }
};
```

Client code:

```
int main()
{
  Record m;
  cout << m.ID << endl;
}
```