CS 32 Worksheet Week 2

Concepts: Copy constructors, assignment operators

1. (5 minutes) What is the output of the following code?

```
#include <iostream>
using namespace std;

class A {
  public:
    A() { cout << "DC" << endl; }

    A(const A& other) { cout << "CC" << endl; }

    A& operator=(const A& other) {
        cout << "AO" << endl;
        return *this;
    }

    ~A() { cout << "Destructor!" << endl; }
};

int main() {
    A arr[3];
    arr[0] = arr[1];
    A x = arr[0];
    x = arr[1];
    A y(arr[2]);
    cout << "DONE" << endl;
}
```

2. (10 minutes) Find the **4 errors** in the following class definitions so the main function runs correctly.

```
#include <iostream>
#include <string>
using namespace std;

class Account {
 public:
   Account(int x) { cash = x; }
   int cash;
}

class Billionaire {
 public:
   Billionaire(string n) {
       offshore = Account(1000000000);
       name = n;
   }

   Account account;
   Account* offshore;
   string name;
};

int main() {
   Billionaire jim = Billionaire("Jimmy");
   cout << jim.name << " has " << jim.account.cash + jim.offshore->cash
        << endl;
}
```

Output:

```
Jimmy has 1000010000
```

3. (10 minutes) What is the output of the following code:

```
#include <iostream>
using namespace std;

class B {
   int m_val;

 public:
   B(int x) : m_val(x) { cout << "Wow such " << x << endl; }
   B(const B& other) {
       cout << "There's another me???" << endl;
       m_val = other.m_val;
   }
   ~B() { cout << "Twas a good life" << endl; }
};

class A {
   int m_count;
   B* m_b;

 public:
   A() : m_count(9.5) {
       cout << "Construct me with " << m_count << endl;
       m_b = new B(m_count + 10);
   }
   A(const A& other) {
       cout << "Copy me" << endl;
       m_count = other.m_count;
       m_b = (other.m_b != nullptr) ? new B(*other.m_b) : nullptr;
   }
   ~A() {
       cout << "Goodbye cruel world" << endl;
       if (m_b) delete m_b;
   }
   int getCount() { return m_count; }
};

int main() {
   A a1, a2;
   A a3 = a2;
   B b1(a3.getCount());
   cout << "Where are we?" << endl;
}
```

4. (15 minutes) Complete the copy constructor, assignment operator, and destructor of the following class. Be careful to avoid aliasing, memory leaks, and other pointer issues!

```
#include <iostream>
using namespace std;

class A {
 public:
   A(int sz) {
       //...implement this!
   }

   A(const A& other) {
       //...implement this!
   }

   A& operator=(const A& other) {
       //...implement this!
   }

   //...other functions

   ~A() {
       //...implement this!
   }

 private:
   B* b;  // one dynamically allocated B object; assume B has a
       // default constructor, a copy constructor, and an
       // assignment operator

   int* arr; // dynamically allocated array

   int n; // size of arr (determined by a constructor)

   string str;
};
```

5. (5 minutes) After being defined by the above code, Jim the Billionaire funded a cloning project and volunteered himself as the first human test subject. Sadly, all his money isn't cloned, so his clone has his name, but has $0. Add the needed function to the Billionaire class so the following main function produces the following output.

```
int main() {
   Billionaire jim = Billionaire("Jimmy");
   Billionaire jimClone = jim;
   cout << jimClone.name << " has "
        << jimClone.account.cash + jimClone.offshore->cash << endl;
   cout << jim.name << " has " << jim.account.cash + jim.offshore->cash
        << endl;
}
```

Output:

```
Jimmy has 0
Jimmy has 1000010000
```