Intermediate Programming Practice Exam 2

1. (10 points) What is your name?

2. (10 points) Please complete the copy constructor in this Java class to make a deep copy.

```
class Stoofs
{
      int x;
      double y;

      Stoofs(Stoofs that)
      {
            x = that.x;
            y = that.y;
      }
}
```

3. (5 points) Please add a copy constructor to this Java class that makes a deep copy.

```
class Jiggy
{
      static int z = 0;
      String s;
      double y;

      Jiggy()
      {
            s = "somestring";
            z++;
            y = 3.14159;
      }

      Jiggy(Jiggy j)
      {
            y = j.y;
            s = new String(j.s);
      }



}
```

4. (10 points) Please complete the copy constructor in this Java class to make a deep copy. (Use the Stoofs copy constructor that you wrote for problem #2 as needed.)

```
class Llama
{
      ArrayList<Stoofs> al;

      Llama()
      {
          al = new ArrayList<Stoofs>();
      }

      void add(Stoofs s)
      {
          al.add(s);
      }

Llama(Llama l)
{
 al = new ArrayList<Stoofs>();
 for(int i = 0; i < l.al.size(); i++)
 {
      add(new Stoofs(l.al.get(i)));
 }
}



}
```

5. (5 points) What will be the output of this Java function if you call f(11)?

```
void f(int n)
{
      if(n > 7)
            f(1);
      System.out.print(n);
      if(n > 4)
            f(n / 2);
      System.out.print(n);
}
```

1111522511

6. (5 points) Version control software, such as Git, enables developers to:

- parse HTML into a document object model.
- track and merge changes made to code.
- protect code from being plagiarized.
- dynamically modify the HTML DOM.
- chat online with other developers.
- resolve concurrency issues.
- translate bytecode into machine-level instructions.
- compile Java code into Javascript.

7. (15 points) For each pair of statements, circle the one that describes the generally better coding practice. (It takes 5 circles to achieve the correct answer.)

- Write your whole program in one or two big methods/procedures/functions.
- Organize your code into many small methods/procedures/functions.

- Test each method immediately after you write it.
- Implement as much of your program as possible, then debug it.

- Avoid making deep copies of large objects unless you have a reason.
- Always make a deep copy of an object before you modify it.

- Use member variables whenever possible. Use local variables only when needed.
- Use local variables whenever possible. Use member variables only when needed.

- Design your methods to work no matter what values are passed to them.
- Start each method with some code that checks the parameters, and throws an exception if any unexpected values are found.

8. (15 points) On the back of this page, write a Java class named Vehicle. Write a class named Tank that extends Vehicle. Add a member variable to Tank that stores the quantity of ammo in the Tank. Write a class named Bike that extends Vehicle. Add a member variable to Bike that stores its color as a String. Add a static method to the Tank class named "charge" that accepts an ArrayList<Vehicle> as a parameter. Using polymorphism, implement this method to decrement the ammo in each tank, and change the color of each bike to "red". (Do not use "instanceof".) Return the number of tanks in the ArrayList. You don't need to allocate the ArrayList.

```
abstract class Vehicle
{
      abstract int upd();
}

class Tank extends Vehicle
{
      int ammo;

      int upd()
      {
            ammo--;
            return 1;
      }

      static int 
charge(ArrayList<Vehicle> al)
      {
            int c = 0;
            for(int i = 0; i < 
al.size(); i++)
            {
                  Vehicle v = 
al.get(i);
                  c += v.upd();
            }
            return c;
      }

}

class Bike extends Vehicle
{
      String color;

      int upd()
      {
            color = "red";
            return 0;
      }
}
```

9. (5 points) What would be the output of this code?

```
class Alpha extends Object
{
  String epsilon(String s)
  {
      System.out.print("Q");
      if(s.equals("U"))
         throw new RuntimeException("I");
      return "C";
  }
}

class Beta extends Alpha {
   Beta(String s) throws Exception {
      System.out.print("H");
      System.out.print(epsilon("E"));
   }

   void delta() {
      System.out.print("K");
   }
}

class Gamma {
   Gamma(int w) {
      System.out.print("F");
   }

   public static void main(String[] args) {
      try{
         Beta b = new Beta("T");
         b.delta();
      } catch(Exception e) {
         System.out.print("O");
      }
      System.out.print("X");
   }
}
```

HQCKX

10. (10 points) Please insert synchronization blocks into the following Java code to make it thread-safe (but don't synchronize more code than is necessary). Assume that only one instance of this class exists, but there can be any number of threads that call any of its methods in parallel.

```
class BankAccount {

   private String owner;
   private String address;
   private double balance;


   private Object lock1;
   private Object lock2;
   private Object lock3;
   BankAccount() {


      lock1 = new Object();
      lock2 = new Object();
      lock3 = new Object();


   }

   void deposit(double amount) {


     if(amount < 0) throw new
       RuntimeException("huh?");
      Synchronized(lock1) {

     double tmp = this.balance;


     tmp += amount;


     this.balance = tmp;

      }
   }

   double getCurrentBalance() {

synchronized(lock1) {
     return this.balance;
}

   }
   void withdraw(double amount) {


     if(amount < 0) throw new
       RuntimeException("huh?");

      Synchronized(lock1) {
     this.balance -= amount;
      }

   }
   void changeOwner(String newName,
      String newAddress) {

Synchreonided(lock2)
{
     this.owner = newName;


     this.address = newAddress;
}

   }
}
```

11. (5 points) What do you need to do to grant everyone a legal right to use code that you post on the web?

- Remove any notices containing "Copyright (C)" from the code.
- Require a log-in form to access the code, but display the password to everyone.
- Make your code proprietary.
- Nothing. If it's on the web, the law assumes you intended people to use it.
- Add a license to the code that grants the legal rights you want them to have.
- Add a click-through End User License Agreement to the code.Submit a form to the Library of Congress indicating your intent to grant rights to copy.

12. (5 points) What is likely to occur if you synchronize (or lock) large portions of code that do not really need to be locked?

- A compiler error.
- A race condition.
- A runtime exception.
- Incorrect computations.
- Threads will run concurrently.
- A thread may starve.